

Learning-based Pose Estimation  
with Multimodal Sensing  
and Physics-based Modeling  
for Practical Human Motion Capture

March 2026

Ryosuke Hori

A Thesis for the Degree of Ph.D. in Engineering

**Learning-based Pose Estimation  
with Multimodal Sensing  
and Physics-based Modeling  
for Practical Human Motion Capture**

March 2026

Graduate School of Science and Technology,  
Keio University

Ryosuke Hori



## Abstract

Motion capture (MoCap), which digitizes human movement, is becoming increasingly important as a fundamental technology in many fields such as sports science, rehabilitation, and robotics. However, conventional MoCap systems require multiple cameras and involve substantial constraints related to installation environments and equipment costs, making it difficult to deploy them widely in daily living spaces. This study aims to enable accurate and reliable human motion capture in everyday environments and systematically presents four approaches that utilize diverse sensor modalities.

The first approach addresses the limited field of view and occlusion issues of fixed-camera systems by employing a wrist-mounted omnidirectional camera. A machine learning model that leverages human silhouette images enables robust full-body 3D pose estimation even in daily environments. The second approach enhances robustness under challenging lighting conditions by using an event camera that asynchronously captures brightness changes. This method achieves accurate estimation of body pose and shape in situations where conventional RGB cameras fail, such as low-light environments or rapid movements, and demonstrates the effectiveness of event-based data. The third approach focuses on convenient gait capture in daily life and proposes a framework for estimating full-body walking posture using only a small number of inertial measurement units (IMUs) embedded in shoes. A minimal sensor configuration combined with a mechanism that suppresses error accumulation during long-duration measurements enhances practicality for everyday gait analysis and long-term monitoring. The fourth approach integrates wearable sensor measurements with physics-based simulation to achieve physically consistent MoCap. Using kinematic and dynamic information from smartwatches and smart insoles, a torque-driven human model in a physics simulator reconstructs physically plausible movement.

These approaches independently address issues such as field-of-view constraints, lighting variability, sensor intrusiveness, and physical consistency, while their sensor configurations and estimation principles complement one another. Combining them according to situational needs may achieve a higher level of accuracy and convenience required for MoCap in daily environments. This potential supports monitoring and functional assessment in industrial, medical, and caregiving settings and contributes to the development of advanced frameworks such as human digital twins and next-generation robotics.

# Acknowledgments

The years I spent in the Ph.D. program have been transformative for me, both academically and personally. I was able to complete this journey thanks to the warm encouragement and support of many professors, colleagues, friends, and my family.

**Advisor.** I would like to express my deepest gratitude to my advisor, Prof. Hideo Saito, for his insightful guidance and constant support. He has provided invaluable direction in research, paper writing, participation in academic conferences, and even in facilitating my visit to Carnegie Mellon University. Over the 6 years since my undergraduate studies at Keio University, his continued mentorship has given me opportunities to collaborate with leading researchers around the world and greatly broaden my perspective as a researcher. I am sincerely grateful for the exceptional environment and learning opportunities he has provided.

**Thesis Committee.** I would also like to express my sincere gratitude to my thesis committee members, Prof. Mariko Isogawa, Prof. Komei Sugiura, and Dr. Mitsunori Tada, for taking the time out of their busy schedules to provide constructive feedback and warm guidance on my research and this thesis. Prof. Isogawa has continuously supported my research since I first began as a fourth-year undergraduate student, offering guidance from diverse perspectives over many years. Prof. Sugiura has provided valuable opportunities and insightful advice through his leadership in the Keio-CMU partnership, which greatly enriched my research experience during my stay at CMU. Dr. Tada welcomed me as a Research Assistant

at the National Institute of Advanced Industrial Science and Technology (AIST), and his extensive expertise as the leader of the Digital Human Research Team has been a tremendous source of guidance and opportunity. I am also deeply grateful for his warm support throughout my time at AIST, which consistently encouraged me and allowed me to pursue my research with confidence. I am deeply grateful for the generous support I have received from all of them.

**Collaborators.** I would also like to express my sincere gratitude to Prof. Kris Kitani, who welcomed me for my visit at Carnegie Mellon University and supervised my research during my stay. His logical and insightful guidance greatly refined the final project of my doctoral studies, and his caring and deeply supportive mentorship made my time at CMU both meaningful and inspiring. During my time at CMU, I was also fortunate to collaborate with outstanding student researchers, including Jyun-Ting Song, Zhengyi (Zen) Luo, Jinkun Cao, and Soyong Shin. The opportunity to work closely with Prof. Kitani and these exceptional colleagues made my days at CMU truly invaluable.

I am also deeply grateful to Dr. Tsubasa Maruyama. He provided thoughtful advice on the research I conducted as a Research Assistant, drawing from his extensive expertise. As the leader of the AIST–CMU collaborative project, he coordinated the overall research direction and created numerous opportunities for me to work with researchers across institutions. This international collaboration would not have been possible without his support, and I sincerely appreciate all he has done.

I am sincerely grateful to Dr. Ryo Hachiuma, who kindly supported me from my very first year in research. His dedication to research and his approach to mentoring junior students have long served as a model that I continue to aspire to. He played a crucial role in laying the foundation of my development as a researcher, and I am profoundly grateful for his guidance.

I would also thank all of those who collaborated with me: Prof. Dan Mikami, Prof. Eni Halilaj, Hiroyuki Deguchi and Wataru Ikeda for having exciting discussions.

**Colleagues and Friends.** These 3 years of my Ph.D. journey became the most fulfilling and transformative period of my life, thanks to the support of wonderful colleagues and my friends. I am truly grateful to all the lab members at Keio University and CMU, as well as the researchers at AIST, for the many enjoyable and meaningful days we shared together. In no particular order, I would like to express my gratitude to: Mana Masuda, Mizuki Obayashi, Yuka Nokihara, Reina Ishikawa, Masashi Hatano, Tomoya Matsubara, Jinhyung (David) Park, Erica Weng, Navyata Sanghvi, Rohan Choudhury, Jenny Nan, Shun Iwase, Yoshihisa Shibata, Hiroki Adachi, Takahiro Maeda, Takuma Yagi, Yu Lei, JungEun (Gracie) Kim, Yichen Peng, Yuto Shibata, Kotaro Amaya, Yasuhiro Nitta, Masa Ogata, Daisuke Ichimura, Suguru Kanoga, Kohta Ito, Yuji Kumano, Ryuto Koike, and many others.

**Family.** Finally, I would like to express my gratitude to my family, who have always been by my side and supported me throughout this journey. I am especially grateful to my parents, who have encouraged every challenge I took on and supported me in countless ways. Above all, I want to express my deepest appreciation to my wife, who has been my closest and greatest source of support. The happiness she brought into my daily life, and the way she stayed by my side through both joyful and difficult moments, have meant more to me than words can say.

# Contents

|                                                              |              |
|--------------------------------------------------------------|--------------|
| <b>Abstract</b>                                              | <b>ii</b>    |
| <b>Acknowledgments</b>                                       | <b>iii</b>   |
| <b>List of Figures</b>                                       | <b>xviii</b> |
| <b>List of Tables</b>                                        | <b>xx</b>    |
| <b>1 Introduction</b>                                        | <b>1</b>     |
| 1.1 Motivation . . . . .                                     | 1            |
| 1.1.1 Human Motion Capture . . . . .                         | 1            |
| 1.1.2 Learning-based Human Pose Estimation . . . . .         | 2            |
| 1.1.3 Challenges of Learning-based Pose Estimation . . . . . | 4            |
| 1.2 Research Goal . . . . .                                  | 5            |
| 1.3 Contributions . . . . .                                  | 6            |
| 1.4 Summary . . . . .                                        | 7            |
| <b>2 Review</b>                                              | <b>9</b>     |
| 2.1 Vision Sensor-based Pose Estimation . . . . .            | 9            |
| 2.2 Non-Vision Sensor-based Pose Estimation . . . . .        | 11           |
| 2.3 Physics-based Motion Modeling . . . . .                  | 13           |
| <b>3 Wearable Camera-based Pose Estimation</b>               | <b>15</b>    |
| 3.1 Background . . . . .                                     | 16           |
| 3.2 Method . . . . .                                         | 19           |
| 3.2.1 Human Pose Estimation Network . . . . .                | 20           |

|          |                                                                                                    |           |
|----------|----------------------------------------------------------------------------------------------------|-----------|
| 3.2.2    | Training Data Synthesis . . . . .                                                                  | 20        |
| 3.2.3    | Silhouetting Process for Inference . . . . .                                                       | 22        |
| 3.2.4    | Pseudo-silhouette Generation for Noise Robustness . . . . .                                        | 23        |
| 3.3      | Experiments . . . . .                                                                              | 25        |
| 3.3.1    | Dataset . . . . .                                                                                  | 25        |
| 3.3.2    | Implementation Details . . . . .                                                                   | 26        |
| 3.3.3    | Evaluation Metric . . . . .                                                                        | 27        |
| 3.3.4    | Experiment 1: Investigating the Effectiveness of Our<br>Silhouette-based Pose Estimation . . . . . | 29        |
| 3.3.5    | Experiment 2: Investigating the Effectiveness of Our<br>Pose Estimation Network . . . . .          | 32        |
| 3.4      | Discussion and Limitations . . . . .                                                               | 34        |
| 3.4.1    | Practical Size and Weight of Sensor Devices . . . . .                                              | 35        |
| 3.4.2    | Versatile and Highly Accurate Pose Estimation . . . . .                                            | 35        |
| 3.4.3    | Real-Time Pose Estimation . . . . .                                                                | 36        |
| 3.5      | Summary . . . . .                                                                                  | 37        |
| <b>4</b> | <b>Event-based Pose Estimation</b>                                                                 | <b>38</b> |
| 4.1      | Background . . . . .                                                                               | 39        |
| 4.2      | Method . . . . .                                                                                   | 45        |
| 4.2.1    | Module Details . . . . .                                                                           | 46        |
| 4.2.2    | Loss Function . . . . .                                                                            | 51        |
| 4.2.3    | EventPointMesh Dataset (EPMD) . . . . .                                                            | 53        |
| 4.3      | Experiments . . . . .                                                                              | 56        |
| 4.3.1    | Experimental Settings . . . . .                                                                    | 56        |
| 4.3.2    | Comparison Against Event-based Methods. . . . .                                                    | 59        |
| 4.3.3    | Comparison Against a Frame-based Method . . . . .                                                  | 65        |
| 4.3.4    | Ablation Study . . . . .                                                                           | 67        |
| 4.3.5    | Evaluation Under Various Input Conditions . . . . .                                                | 69        |
| 4.4      | Discussion and Limitations . . . . .                                                               | 72        |
| 4.4.1    | Failure Cases . . . . .                                                                            | 72        |
| 4.4.2    | Room for Improvement in Event-based HMR . . . . .                                                  | 77        |
| 4.5      | Summary . . . . .                                                                                  | 78        |

|          |                                                      |            |
|----------|------------------------------------------------------|------------|
| <b>5</b> | <b>IMU-based Gait Estimation</b>                     | <b>79</b>  |
| 5.1      | Background . . . . .                                 | 80         |
| 5.2      | Method . . . . .                                     | 83         |
| 5.2.1    | Sensor Configuration . . . . .                       | 85         |
| 5.2.2    | Network Details . . . . .                            | 86         |
| 5.2.3    | Loss Functions . . . . .                             | 90         |
| 5.3      | Experiments . . . . .                                | 96         |
| 5.3.1    | Datasets . . . . .                                   | 96         |
| 5.3.2    | Data Synthesis . . . . .                             | 98         |
| 5.3.3    | Implementation Details . . . . .                     | 100        |
| 5.3.4    | Baseline Methods . . . . .                           | 101        |
| 5.3.5    | Evaluation Metrics . . . . .                         | 102        |
| 5.3.6    | Comparison with Baseline Methods . . . . .           | 105        |
| 5.3.7    | Comparison under Identical IMU Placement . . . . .   | 112        |
| 5.3.8    | Ablation Study . . . . .                             | 113        |
| 5.4      | Discussion and Future Work . . . . .                 | 115        |
| 5.4.1    | Practical Demonstration System . . . . .             | 115        |
| 5.4.2    | Handling Straight-Line Walking Constraints . . . . . | 117        |
| 5.4.3    | Addressing Device and Footwear Variability . . . . . | 119        |
| 5.4.4    | Toward Real-World Gait Analysis . . . . .            | 120        |
| 5.4.5    | Toward Clinical Applications . . . . .               | 121        |
| 5.5      | Summary . . . . .                                    | 122        |
| <b>6</b> | <b>Physics-based Pose Estimation</b>                 | <b>124</b> |
| 6.1      | Background . . . . .                                 | 125        |
| 6.2      | Method . . . . .                                     | 129        |
| 6.2.1    | Input Data . . . . .                                 | 129        |
| 6.2.2    | KinematicsNet . . . . .                              | 130        |
| 6.2.3    | State Difference . . . . .                           | 131        |
| 6.2.4    | DynamicsNet . . . . .                                | 132        |
| 6.2.5    | PRISM Dataset . . . . .                              | 135        |
| 6.3      | Experiments . . . . .                                | 145        |
| 6.3.1    | Implementation Details . . . . .                     | 145        |
| 6.3.2    | Comparison . . . . .                                 | 148        |



---

|          |                                      |            |
|----------|--------------------------------------|------------|
| 6.3.3    | Evaluation . . . . .                 | 153        |
| 6.4      | Summary . . . . .                    | 159        |
| <b>7</b> | <b>Conclusion</b>                    | <b>160</b> |
| 7.1      | Summary . . . . .                    | 160        |
| 7.2      | Limitations . . . . .                | 161        |
| 7.3      | Discussion and Future Work . . . . . | 162        |
| 7.3.1    | Research Directions . . . . .        | 163        |
| 7.3.2    | System Design Perspectives . . . . . | 164        |
|          | <b>Achievements</b>                  | <b>190</b> |

# List of Figures

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1 | The overview of this thesis. Chapters 3 and 4 address the limitations of conventional vision-sensor-based pose estimation by employing a wearable camera and an event camera. Chapter 5 improves practical usability by using shoe-embedded IMUs, and Chapter 6 achieves physically plausible pose estimation by integrating physics simulation. . . . .                                                                                                                                                                                                                                      | 6  |
| 3.1 | Concept of our 3D human pose estimation method using a wrist-mounted 360° camera. (a) Future vision depicting a smartwatch with a 360° camera. The area shown in red represents the camera's field of view; (b) Camera view with a 360° camera (GoPro MAX) mounted on the wrist, which is the camera configuration of our method; (c) 3D pose of the user estimated from (b) image by our 3D human pose estimation framework. . . . .                                                                                                                                                         | 16 |
| 3.2 | Overview of our 3D pose estimation method using a single wrist-mounted 360° camera. The network extracts time-series features from a silhouette image sequence and predicts a 3D human pose sequence. The input data for the network during training is a synthetic silhouette image sequence generated by animating a humanoid using MoCap data as described in Sec. 3.2.2. During inference, the input is a silhouette image sequence obtained by our silhouetting process from equirectangular images captured by a 360° camera in a real environment, as described in Sec. 3.2.3. . . . . | 19 |

|     |                                                                                                                                                                                                                                                                                                                                                                                       |    |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.3 | Overview of training data synthesis. We animated a humanoid avatar using MoCap data and captured equirectangular images with a virtual 360° camera fixed at the avatar’s wrist position. To capture binary silhouette images, we set the model’s body to white and the background to black. . . .                                                                                     | 21 |
| 3.4 | Our silhouetting process, which converts an equirectangular image captured by a wrist-mounted 360° camera into a binary silhouette image. We extract the silhouette of the camera wearer from the region labeled as human, which is estimated by an existing semantic segmentation model. . . .                                                                                       | 22 |
| 3.5 | Training the pix2pix model to map our synthetic silhouette images with smooth contour to noisy silhouette images obtained through our silhouetting process. The discriminator, $D$ , learns to classify between fake (synthesized by the generator) and real {smooth silhouette image, noisy silhouette image} tuples. The generator, $G$ , learns to fool the discriminator. . . . . | 24 |
| 3.6 | Motion capture for constructing our MoCap Training and Test Data. Subjects wore motion capture markers and the 360° camera as shown in Fig. (a) and were asked to perform actions as shown in Fig. (b)-(e). . . . .                                                                                                                                                                   | 26 |
| 3.7 | Examples of training and test data for each method in experiment 1. We used the same silhouette images for training our silhouette-based methods (SS, HRNet, and Swin). . . . .                                                                                                                                                                                                       | 29 |
| 3.8 | Qualitative results for each method on the MoCap test data (left) and the in-the-wild data (right) in Experiment 1. . . . .                                                                                                                                                                                                                                                           | 32 |
| 3.9 | Qualitative results for each method on the MoCap test data (left) and the in-the-wild data (right) in Experiment 2. . . . .                                                                                                                                                                                                                                                           | 35 |
| 4.1 | Event cameras are designed to detect luminance changes, offering high temporal resolution and a high dynamic range. These unique features effectively address the challenges faced by frame-based cameras, such as motion blur and low frame rates in poorly-lit environments. . . . .                                                                                                | 40 |

- 4.2 We propose EventPointMesh, a method for 3D human mesh recovery solely from event data. EventPointMesh treats event data as point clouds and estimates human meshes from these point clouds segmented by fixed-time intervals. It enables high-speed HMR that is unaffected by lighting conditions. . . . . 41
- 4.3 Pipeline of our event-based HMR method, EventPointMesh. It consists of four modules: the Base Module, Keypoints Module, Anchor Points, and SMPL Module (Sec. 4.2.1). To achieve highly accurate mesh reconstruction, EventPointMesh employs a coarse-to-fine feature extraction process: the Keypoints Module extracts global features from the event point cloud, and the AnchorPointsModule extracts local features. . . . . 44
- 4.4 Detailed illustrations of the spatiotemporal grouping of event point clouds in the Anchor Points Module. Yellow points represent the 2D joint positions estimated by the Keypoints Module, while red and blue indicate the input event point clouds, and green represents the grouped points. (a) An example of applying spatiotemporal grouping to real event data, focusing on the right wrist, though in practice, this process is applied individually to all major joints. (b) A detailed view of the nearest points selection. For each point in the event point clouds scattered in space and time, the L2 norm on the image plane with the 2D joint coordinates is calculated, and the nearest  $G$  points with the closest distances are selected as a group. . . . . 49
- 4.5 The EventPointMesh Dataset (EPMD) encompasses diverse actions from daily activities to sports motions. Subjects performed the actions multiple times in both well-lit and poorly-lit environments. . . . . 54

- 4.6 Qualitative results from the experiment using MMHPSD as detailed in Sec 4.3.2. The top rows display the results under the all-subject setting, while the bottom rows show the results under the cross-subject setting. Although our proposed method only uses event data as input, it estimates the mesh with an accuracy comparable to EventHPE(VIBE), which uses intensity images to estimate the initial mesh for each sequence, and EventHPE(GT), which employs ground truth mesh. . . . . 60
- 4.7 Qualitative results from the experiment using EPMD under the all-subject setting (Sec 4.3.2). The top two rows display well-lit data results, while the bottom two rows show poorly-lit data results. The images, segmented into four blocks separated by wide spaces, are adjacent frames arranged in the top and bottom rows. Note that the Optical Flow, assisting mesh estimation in EventHPE(VIBE) and EventHPE(GT), comes from the FlowNet module and is unrelated to EMP (Ours). . . . . 63
- 4.8 Qualitative results of the experiment comparing our event-based method with the frame-based method. The top row displays the mesh estimation results of both methods in bright environments, while the bottom row presents the estimation results in dark environments. . . . . 66
- 4.9 Qualitative results of the Task2 described in Sec. 4.3.5. The topmost “EPM(img)” represents our method’s results when segmenting events according to the intensity image frame rate, which varies with the lighting conditions. The middle “EPM(120)” shows the outcomes when segmenting events and estimating the meshes at the same 120fps. The bottom “GT” shows the ground truth meshes with a frame rate of 120fps. Note that the transparent meshes in the bottom two rows are only for visual clarity; there is no difference from the opaque meshes. . . . . 69

- 4.10 Failure cases of mesh estimation by EventPointMesh. The top row presents the output of our method for a scenario discussed in 4.4.1 Case 1, where the subject steps out of the camera’s view. Currently, in cases where no events are generated due to the absence of the subject, a mesh in a typical standing pose is outputted. The bottom row shows an example of mesh estimation when events are generated by parts other than the subject, as discussed in 4.4.1 Case 2. It demonstrates that the presence of shadows causing events can lead to a reduction in mesh estimation accuracy. . . . . 75
- 5.1 Overview of the proposed Gait Inertial Poser (GIP). (a) User wearing minimal inertial measurement units (IMUs) attached only to shoe soles, enabling unobtrusive daily gait measurement. (b) Input data consisting of easily obtainable user attributes (height, weight, age, gender) and IMU measurements (acceleration and angular velocity). These inputs allow estimation of the user’s personalized body model and gait parameters, including gait phase, velocity, and joint angles. (c) Final output visualizing smooth, accurate full-body walking motions tailored to the user’s characteristics. . . . . 80
- 5.2 Detailed network architecture of the proposed method. (a) Input includes user attributes (height, weight, age, gender) and IMU measurements (acceleration, angular velocity). (b) The Body Module estimates personalized body shape parameters. (c) The Gait Module, based on a Transformer, simultaneously estimates gait phase, joint angles, velocities, root positions, and foot trajectories using IMU data. (d) The Smoothing Module employs a Variational Autoencoder (VAE) to enhance temporal smoothness and stability of the estimated motion, effectively reducing IMU drift through loss functions inspired by zero velocity updates (ZUPT). (e) Final output is a smooth, stable full-body walking motion reconstruction. . . . . 84

- 5.3 Example of the gait motion data from the AIST Gait Database. Subjects walked naturally along a straight path, stepping on force plates (indicated in blue), allowing simultaneous capture of motion and ground reaction force (GRF) data (indicated in red and green). The GRF data were used to define gait phases such as heel-strike and toe-off. . . . . 96
- 5.4 Synthetic IMU sensor positions defined on SMPL mesh vertices (blue labels indicate SMPL vertex IDs). The foot soles correspond to the proposed method, while positions on the left wrist, left ear, and right thigh simulate IMU signals for baseline methods (smartwatch, earphone, smartphone in pocket). The yellow arrows indicate vectors used to define the mesh face normal for establishing IMU coordinate systems. . . . . 99
- 5.5 Qualitative comparison of foot trajectory (left) and full-body pose estimation (right) results using the AIST Gait DB. The results show side and top views for baseline methods (Integration, Integration + ZUPT, IMUPoser, MobilePoser) and Ours, compared with the Ground Truth. . . . . 108
- 5.6 Qualitative comparison of foot trajectory estimation results on the UnderPressure. The side and top views illustrate estimated foot trajectories for Integration, Integration + ZUPT, Ours, and the Ground Truth. . . . . 109
- 5.7 Qualitative comparison of full-body pose estimation results on the UnderPressure. Side and top views show results from IMUPoser, MobilePoser, Ours, and the Ground Truth. Compared to baseline methods, our approach achieves more accurate and natural full-body pose reconstructions, demonstrating stable performance even with real IMU data. . . . . 110

- 5.8 Visualization of joint position error (PEL-MPJPE) distributions. **Top:** Heatmaps illustrating that joints closer to IMU attachment sites (indicated by pink stars) exhibit lower positional errors. **Bottom:** Box plots detailing joint position errors for different body segments. Our method, utilizing IMUs attached to the soles of both feet, achieves notably higher accuracy in the lower limbs, essential for gait analysis. . . . . 111
- 5.9 Foot Velocity Error of estimated poses over two consecutive windows, comparing the Gait Module (blue) and Smoothing Module (red). The results are computed on the Under-Pressure dataset. . . . . 115
- 5.10 Demonstration system of our proposed method using consumer-grade IMU devices (ORPHE CORE). IMU data captured by sensors embedded in shoe soles are transmitted either directly to a laptop computer via Bluetooth (blue arrow), or relayed through a smartphone (magenta arrows) to minimize BLE interference. This setup ensures reliable gait pose estimation in practical scenarios. . . . . 116
- 5.11 Example of straight-walking segmentation in the UnderPressure dataset. Blue indicates straight-walking segments and red indicates non-straight segments. Only the straight-walking segments were used for training and evaluation of the proposed method. . . . . 118
- 6.1 Overview of the proposed Ground Reaction Inertial Poser (GRIP). (a) GRIP observes motion using four IMUs and foot pressure data from smartwatches and smart insoles. (b) Full-body motion is reconstructed by driving a humanoid with joint torques in a physics simulator. (c) The PRISM dataset offers multimodal measurements, including IMUs, foot pressure, motion data, and environmental data. . . . . 125



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                              |     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.2  | Overview of the GRIP framework. <b>Input Data</b> (Sec. 6.2.1) consists of IMU and insole measurements. <b>KinematicsNet</b> (Sec. 6.2.2) estimates kinematic states, and the <b>State Difference</b> (Sec. 6.2.3) compares them with the simulated humanoid. <b>DynamicsNet</b> (Sec. 6.2.4) drives the humanoid through physics simulation-based control. The <b>PRISM dataset</b> (Sec. 6.2.5) provides diverse multi-modal data. . . . . | 128 |
| 6.3  | Calibration setup for initial orientation. Left: Apple Watch. Right: Xsens MTw sensors. . . . .                                                                                                                                                                                                                                                                                                                                              | 137 |
| 6.4  | Sensor and global coordinate frames for the OpenGo devices, illustrating the handedness difference between sensors.                                                                                                                                                                                                                                                                                                                          | 139 |
| 6.5  | Alignment of the Aria Glass SLAM trajectory to the MoCap-based head trajectory using Procrustes analysis. . . . .                                                                                                                                                                                                                                                                                                                            | 140 |
| 6.6  | Vertical acceleration signals from MoCap data and from IMU sensors at the same set of body-mounted locations. The characteristic jump-induced peaks provide reliable cues for temporal synchronization across sensor streams. . . . .                                                                                                                                                                                                        | 141 |
| 6.7  | Qualitative comparison of pose estimation results across the three datasets. Our method accurately reconstructs foot placement on objects (PRISM), exhibits less position drift (UnderPressure), and captures slow weight-shifting motions (PSU-TMM100). . . . .                                                                                                                                                                             | 149 |
| 6.8  | Comparison of foot contact timing. Right-foot contact labels during low-speed motions in PSU-TMM100, computed from the estimated GRF of the physics-based methods. . . .                                                                                                                                                                                                                                                                     | 150 |
| 6.9  | Qualitative comparison of estimated poses and root trajectories for a walking sequence from the UnderPressure dataset. Colors correspond to the same methods shown in Fig. 6.7. .                                                                                                                                                                                                                                                            | 151 |
| 6.10 | Comparison between SolePoser and our method using the same 17 joint locations extracted from the SMPL model. . . .                                                                                                                                                                                                                                                                                                                           | 152 |
| 6.11 | Success rates for different motion categories. Lower success rates are observed for balance motions, fast movements, and object interaction tasks compared to daily motions. . . . .                                                                                                                                                                                                                                                         | 157 |

---

|                                                                                                                                                                                                               |     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.12 Temporal evolution of translation error over a sequence. The translation error increases approximately linearly with time, reflecting the accumulation of drift in absolute position estimation. . . . . | 158 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|

# List of Tables

|     |                                                                                                                                                                                                                                                                       |     |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.1 | Quantitative results of evaluating the accuracy of pose estimation on the MoCap test data and the in-the-wild data in Experiment 1. Bold values show the best scores. . . . .                                                                                         | 31  |
| 3.2 | Quantitative evaluation results of pose estimation accuracy and joint acceleration on the MoCap test data and the in-the-wild data in Experiment 2. . . . .                                                                                                           | 34  |
| 4.1 | Datasets for event-based human pose and shape estimation                                                                                                                                                                                                              | 54  |
| 4.2 | Quantitative results from experiments using the MMHPSD and EPMD datasets. . . . .                                                                                                                                                                                     | 59  |
| 4.3 | Quantitative Results of the Comparison between Event-based and Frame-based Methods. . . . .                                                                                                                                                                           | 65  |
| 4.4 | Quantitative results from the ablation study . . . . .                                                                                                                                                                                                                | 68  |
| 4.5 | Quantitative evaluation under various input conditions. . . .                                                                                                                                                                                                         | 70  |
| 5.1 | Performance comparison of foot trajectory and pose estimation methods across different datasets (AIST Gait Database and UnderPressure Dataset) and evaluation metrics. Bold values indicate the highest accuracy achieved by each metric within each dataset. . . . . | 106 |
| 5.2 | Comparison under identical IMU placement (two IMUs on foot soles). Bold indicates the best-performing method for each metric. . . . .                                                                                                                                 | 113 |
| 5.3 | Ablation study of the proposed method, where B, G, and S denote the Body, Gait, and Smoothing Modules, respectively.                                                                                                                                                  | 114 |

---

|     |                                                                                                                                                                                                                                                                   |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.4 | Quantitative comparison of estimation accuracy across various motions, including straight walking at multiple speeds (Normal, Slow, Fast), non-straight walking, and jogging in the UnderPressure dataset. . . . .                                                | 117 |
| 6.1 | Comparison of human motion datasets utilizing insole-type foot pressure sensors. . . . .                                                                                                                                                                          | 143 |
| 6.2 | Quantitative comparison of our method with baseline methods across the three datasets. Lower values indicate better performance for all metrics. Bold numbers denote the best performance for each metric, and underlined numbers denote the second best. . . . . | 147 |
| 6.3 | Comparison of input modalities, output representations, and physical modeling across baseline methods. . . . .                                                                                                                                                    | 148 |
| 6.4 | Comparison across different sensor configurations. . . . .                                                                                                                                                                                                        | 154 |
| 6.5 | Comparison with and without the height map. . . . .                                                                                                                                                                                                               | 155 |
| 6.6 | Ablation study on kinematic estimation used in DynamicsNet.                                                                                                                                                                                                       | 156 |
| 6.7 | Comparison on different sequence lengths. . . . .                                                                                                                                                                                                                 | 158 |

# Chapter 1

## Introduction

### 1.1 Motivation

#### 1.1.1 Human Motion Capture

Motion capture (MoCap), a technology that digitizes human movements, plays a fundamental role in a wide range of fields, including sports analysis, medical rehabilitation, robotics, VR/AR, and industrial support. If MoCap could be easily deployed in everyday environments, it would enable new applications that are difficult to realize in controlled laboratory settings, such as long-term monitoring of gait and posture at home, remote assessment of rehabilitation progress, context-aware support in work environments, and the construction of personalized digital twins. For these reasons, achieving practical MoCap in daily living spaces carries significant academic and societal value.

Marker-based optical MoCap (e.g., OptiTrack [99], Vicon [82]), which tracks reflective markers attached to the body using infrared cameras, and markerless multi-view MoCap (e.g., The Captury [23], Theia3D [50]), which estimates human pose using multi-view RGB cameras, have been widely

used. Although these systems can provide high-precision measurements, they require strict conditions such as multiple calibrated cameras, dedicated capture spaces, and appropriate lighting, making them difficult to deploy in everyday environments. A different line of approaches employs wearable MoCap systems using Inertial Measurement Units (IMUs), with representative commercial systems including Xsens [49], Rokoko [4], and Noitom [81]. These systems require attaching many IMUs to various body parts, resulting in substantial physical burden and setup complexity, which remain major obstacles to everyday use. Consequently, there is an increasing demand for MoCap methods that are easier to use and less sensitive to environmental conditions, and recent research has increasingly focused on deep learning based pose estimation techniques.

The main reason deep learning has attracted significant attention is its ability to infer human pose even when observations are incomplete. Traditional optical or inertial MoCap methods are prone to failure when observations become unreliable, for example when the subject moves out of view, markers become occluded, or inertial sensors drift. In contrast, deep learning models leverage statistical structures learned from large datasets, such as human body geometry and typical motion patterns, allowing them to compensate for missing observations and still estimate the full-body state. This property is particularly beneficial when aiming for practical MoCap systems that must operate under device limitations and varying real-world conditions, rather than in controlled studio settings.

### **1.1.2 Learning-based Human Pose Estimation**

With the rapid advancement of deep learning, a wide range of learning-based approaches have been developed to estimate human pose and motion from sensor data. These methods take measurements from various sensors as input and use machine learning models to estimate a human

body model that reproduces the state of a real person. Human body models can be broadly categorized into kinematic models, which represent joint positions, joint angles, and full-body 3D meshes, and dynamic models, which incorporate physical quantities such as body mass, inertial properties, joint torques, and external forces including ground reaction forces. Most existing methods focus primarily on estimating kinematic models since kinematic models rely only on observable quantities, whereas dynamic models require ground-truth forces and torques that are expensive and difficult to acquire.

A diverse range of sensor modalities has been explored, including vision sensors (e.g., RGB, depth, infrared, event), wireless signal sensors (e.g., WiFi, millimeter-wave radar, RFID), sound-based sensors (e.g., ultrasonic and other acoustic signals), and wearable sensors (e.g., inertial measurement units, electromagnetic sensors, head-mounted displays). Machine learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based architectures have been widely used, and capturing temporal dependencies remains a critical component of reliable pose estimation.

Learning-based MoCap approaches require pairs of motion data and synchronized sensor measurements. Large-scale datasets that encompass diverse sensor modalities, human motions, and environmental conditions play an essential role in advancing the field, as they enable systematic development, comparison, and evaluation of different models. However, acquiring high-precision reference motion using marker-based MoCap and collecting synchronized multi-sensor measurements is costly, and it is not easy to comprehensively capture a wide range of environments and motion types. As a result, the high cost of data acquisition remains a major challenge in learning-based MoCap, and the development of data synthesis methods as well as the construction of large-scale synthetic datasets has become one of the important research topics for addressing this limitation.

### 1.1.3 Challenges of Learning-based Pose Estimation

Conventional learning-based pose estimation methods face several important challenges. First, widely used camera-based approaches are highly sensitive to environmental factors such as limited camera viewpoints, occlusions, and variations in illumination. As a result, it is difficult to achieve stable estimation in uncontrolled environments such as homes, medical and nursing-care facilities, and industrial sites, where object layouts are complex and lighting conditions can vary significantly.

Second, methods that rely on wearable devices suffer from issues related to the complexity of sensor attachment, physical burden on the user, and discomfort during long-term use. Although increasing the number of sensors can improve estimation stability, doing so reduces the convenience and portability of the system, making it difficult for general users to operate. Therefore, methods capable of achieving high-accuracy estimation with a minimal number of sensors are required.

Third, the kinematics-based pose estimation widely adopted in existing methods can produce physically unrealistic postures and violations of contact constraints, including foot sliding, penetration, and floating. This issue becomes particularly evident in highly dynamic movements or in scenarios involving interactions with the environment or objects, undermining the reliability of the estimated motion.

Overcoming these combined challenges, including vulnerability to environmental conditions, complexity of sensor attachment, and the lack of physical consistency, is essential for establishing MoCap techniques that can operate robustly in everyday environments. This study aims to address these issues from multiple perspectives by developing and evaluating several approaches that leverage diverse sensor modalities and physics-based modeling.



## 1.2 Research Goal

To achieve practical and reliable learning-based human motion capture in everyday environments, it is essential to systematically overcome several fundamental limitations inherent in conventional approaches. To guide this work, we formulate the following three core research questions (RQs):

- **RQ1:** How can we mitigate the vulnerability of widely used vision-based methods to environmental factors, including limited viewpoints, occlusions, and illumination changes?
- **RQ2:** How can we achieve accurate pose estimation with a minimal number of wearable sensors, without imposing excessive constraints on sensor attachment and while maintaining practical usability for everyday users?
- **RQ3:** How should data-driven models and physics-based modeling be integrated to ensure that the reconstructed motions satisfy contact constraints and dynamic consistency, yielding physically plausible human motion?

This thesis explores multiple approaches based on different sensing modalities and estimation frameworks to address these three questions from complementary perspectives. Specifically, we investigate vision-based approaches using wearable cameras to alleviate issues related to limited viewpoints and occlusions through egocentric vision, as well as event cameras to enhance robustness to illumination changes. To improve practicality and reduce the burden of sensor attachment, we examine gait estimation using a shoe-embedded IMU. Furthermore, we propose an approach that combines smartwatch and insole sensors, which are non-vision modalities, with physics simulation to estimate full-body motions with physical consistency even under sparse sensing.

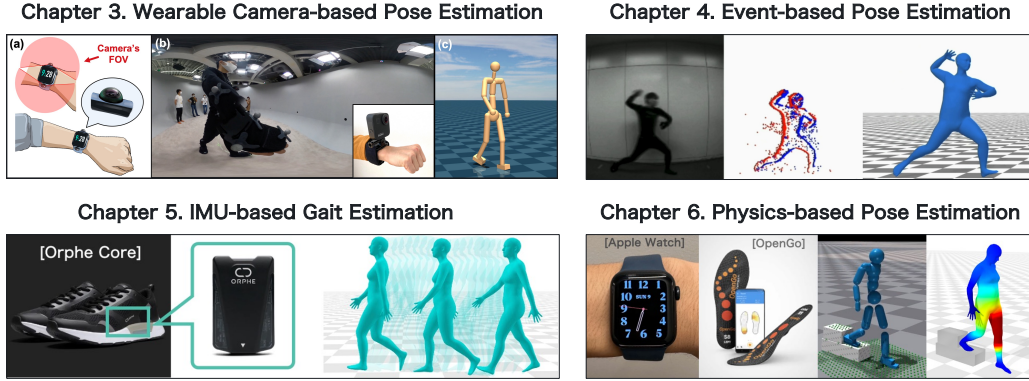


Figure 1.1: The overview of this thesis. Chapters 3 and 4 address the limitations of conventional vision-sensor-based pose estimation by employing a wearable camera and an event camera. Chapter 5 improves practical usability by using shoe-embedded IMUs, and Chapter 6 achieves physically plausible pose estimation by integrating physics simulation.

Each approach primarily contributes to one or more of the above research questions, and their detailed relationships and contributions are summarized in the next section. The overarching goal of this thesis is to address the challenges associated with environmental vulnerability, constraints on sensor attachment, and physical inconsistency, thereby establishing foundational techniques for reliable learning-based human motion capture in everyday environments.

### 1.3 Contributions

The main contributions of this thesis are as follows.

- **Chapter 3** introduces a wearable camera-based pose estimation method using a wrist-mounted 360-degree camera, addressing the issues of limited viewpoints and occlusions inherent in conventional third-person camera systems. This chapter primarily contributes to **RQ1**.

In addition, by relying on a simple setup with a single device, it also contributes to **RQ2**.

- **Chapter 4** presents a method for reconstructing human meshes solely from event data, without relying on frame-based intensity images, thereby improving robustness to illumination changes. This provides an additional solution to the environmental challenges faced by vision-based methods and further supports **RQ1**.
- **Chapter 5** proposes a gait estimation method that uses only shoe-embedded IMU sensors, significantly reducing attachment burden while maintaining practical estimation accuracy. This directly contributes to **RQ2**, as it achieves accurate pose estimation with minimal constraints on sensor attachment.
- **Chapter 6** demonstrates a method for reconstructing physically plausible full-body motion by combining smartwatches and smart insoles with physics simulation. While this chapter primarily contributes to **RQ3** by enhancing physical plausibility, it also contributes to **RQ2**, as the method operates with only two types of devices, four units in total.

## 1.4 Summary

In this chapter, we outlined the background and challenges involved in achieving learning-based human motion capture that functions reliably in everyday environments, and we presented the three fundamental research questions (RQs) addressed in this thesis. We also described the overall approach of this work, which investigates multiple methods that combine diverse sensing modalities with physics-based modeling to respond to these questions.

---

The following chapters are organized as follows. Chapter 2 provides a comprehensive review of related works on vision-based pose estimation, non-vision sensor-based pose estimation, and physics-based motion modeling. Chapter 3 introduces a pose estimation method based on a wearable camera. Chapter 4 presents a mesh reconstruction approach using an event camera. Chapter 5 investigates a gait estimation method using shoe-embedded IMUs. Chapter 6 introduces a full-body motion estimation method that integrates sparse non-vision sensors with physics simulation. Finally, Chapter 7 summarizes the overall contributions of this thesis and discusses future directions.

# Chapter 2

## Review

In this chapter, we provide an overview of related work on learning-based human motion capture, which forms the foundation of this thesis. Technologies for accurately measuring full-body human motion have historically been developed around commercial systems such as optical motion capture [82,99] and multi-view markerless approaches [23,50]. While these systems offer well-established accuracy, they require dedicated studios with significant installation and maintenance costs, involve substantial effort for multi-camera calibration and marker attachment, and are difficult to operate in ordinary living environments. These limitations have led to growing interest in machine learning based approaches that can function more flexibly in daily settings. In this chapter, we review the overview of relevant techniques from three perspectives: vision sensors, non-vision sensors, and physics-based modeling.

### 2.1 Vision Sensor-based Pose Estimation

Vision-based three-dimensional human pose estimation (3D HPE) has been an active research area for many years in the computer vision community,

driven by its broad applicability in fields such as animation, virtual and augmented reality, healthcare, and sports [20, 77, 113]. In particular, RGB camera based third-person methods, including both monocular and multi-view approaches, have advanced significantly with the development of deep learning techniques [61, 91, 92, 101, 102, 126, 133]. While these methods have demonstrated strong performance in everyday visual environments, their reliance on externally positioned cameras inherently limits the usable motion range and makes them vulnerable to occlusions and field-of-view constraints.

To address these limitations, egocentric approaches that utilize body-mounted cameras have been explored. Early work primarily focused on recognizing activities or estimating upper-body posture from wearable cameras [7, 28, 110, 160]. Subsequent research expanded toward full-body estimation by attaching multiple wearable cameras or leveraging wide-angle lenses [15, 16, 108, 124]. More recently, considerable efforts have been devoted to achieving whole-body pose estimation using only a single egocentric camera, including inside-out settings where the wearer's body is barely visible [53, 54, 100, 161, 162] and configurations using fisheye lenses to capture a wider field of view [44, 48, 131, 132, 140, 150]. These egocentric approaches alleviate the constraints of third-person systems by enabling wide-range activity capture and robustness to occlusions, while still facing challenges such as limited visible body area or dependence on specialized lens configurations.

In recent years, event cameras have emerged as an alternative vision modality for human pose and mesh estimation. Unlike conventional RGB cameras that record intensity images at fixed frame rates, event cameras asynchronously capture per-pixel luminance changes, resulting in extremely high temporal resolution, low latency, and strong robustness to low-light conditions. Research on event-based HPE and HMR has explored a variety of event representations, including event frames [10], tensor-based

aggregations [114], the TORE representation [6, 169], and point-based representations [18]. Further advances have incorporated temporal modeling to improve 2D and 3D joint estimation [33, 120], as well as parametric body modeling for event-based mesh reconstruction [111, 148, 175]. Owing to their resilience to illumination changes, low computational cost, and potential for privacy-preserving motion capture, event cameras have gained attention as a promising complement or alternative to RGB-based methods.

In summary, vision-based pose estimation has evolved along three primary directions: third-person estimation with externally placed RGB cameras, egocentric estimation with body-mounted cameras, and event-based estimation using neuromorphic vision sensors. Despite their progress, these methods remain influenced by environmental factors such as occlusion, field-of-view limitations, and lighting conditions. The following chapters of this thesis explore vision-based approaches that address these challenges through wearable camera setups and event-based sensing.

## 2.2 Non-Vision Sensor-based Pose Estimation

Unlike vision-based methods, non-vision sensing approaches are less affected by lighting conditions or occlusions and therefore offer a promising alternative for motion estimation in everyday environments. In particular, IMUs, foot pressure sensors, and RF signals can capture human motion and human–environment interactions without relying on visual information, and a wide range of research has explored their potential.

First, full-body pose estimation using a small number of IMUs has been extensively studied. Early work showed that accurate motion reconstruction is feasible even with sparse IMUs through optimization-based approaches [138] or deep learning [46]. Subsequent studies improved global

localization accuracy and physical consistency by incorporating additional cues such as ground–contact reasoning [156], physics constraints [155], terrain estimation [59], non-inertial effects [157], and interactions with external objects [153]. Other methods integrate auxiliary sensing modalities such as UWB ranging [5, 75], barometric altitude [165], or egocentric vision [34, 36, 70, 154, 158]. Recent studies have explored lightweight configurations using everyday devices [42, 95, 149, 176], as well as generative models that synthesize motions for uninstrumented body parts [137]. However, reducing the number of sensors inherently limits the complexity of motions that can be estimated and makes it difficult to capture body–environment interactions, suggesting that IMUs alone have fundamental limitations.

Foot pressure sensors provide valuable information about human motion dynamics and foot–ground interactions. Studies using fixed pressure mats have demonstrated improvements in physical plausibility and pose estimation accuracy [19, 83, 106, 134], but their applicability is restricted to controlled environments. Portable insole-type sensors have been used to analyze stability [117], support vision-based pose estimation [164], and reduce foot-skating artifacts [96]. Moreover, IMU–pressure fusion approaches have been proposed to estimate root-relative body poses [145] or reconstruct full-body posture and global trajectories using dual-modal input [159]. Nevertheless, foot-only configurations cannot fully capture upper-body motion, and IMUs operating in local sensor coordinates make global trajectory estimation difficult. Existing methods also lack mechanisms for enforcing physical plausibility. These issues highlight the need for complementary information and stronger modeling frameworks.

In addition, RF-based pose and shape estimation methods have emerged as promising alternatives. WiFi and millimeter-wave signals are resilient to low-light conditions and can penetrate obstacles, enabling estimation in scenarios where RGB(D) cameras struggle [3, 58, 73, 107, 118, 151, 171]. However, RF signals have inherently low spatial resolution and are strongly



affected by multipath interference and environmental variability, making reliable full-body estimation difficult in real-world settings.

Overall, non-vision sensing modalities offer powerful ways to alleviate the environmental limitations of vision-based methods, but they also introduce their own constraints stemming from sensor placement and modality characteristics. Building on these observations, this thesis explores approaches that utilize sparse IMUs and insole-type pressure sensors, and integrates physics simulation where necessary, to achieve accurate and physically consistent full-body motion reconstruction with minimal wearable devices.

## 2.3 Physics-based Motion Modeling

Learning-based human motion capture methods are typically trained to reproduce kinematic quantities such as joint angles and pose parameters. However, because these approaches do not explicitly enforce physical laws, the estimated motions often exhibit physical inconsistencies, including foot sliding, ground penetration, and unstable body balance. To address these issues, a wide range of physics-based approaches has been explored to enhance the physical plausibility of estimated motions. Existing work on physics-based modeling can be broadly categorized into three groups: (i) physics-based optimization, which refines estimated motions through post-hoc optimization; (ii) physics-based simulation, which reconstructs poses by controlling a virtual human in a physics engine; and (iii) physics-integrated learning models, which embed physical models or dynamics-based losses directly into the neural network.

In the physics-based optimization paradigm, many image-based methods refine initial pose estimates by applying trajectory optimization that incorporates equations of motion and contact constraints, thereby recon-

structing temporally consistent and physically plausible full-body motion [37, 38, 89, 105, 147, 166]. These approaches recover dynamic consistency and stable contact even when the observations are incomplete. Similarly, in IMU-based approaches, kinematic poses are first estimated from sensor measurements and subsequently refined by imposing contact and dynamic constraints, which helps restore physical plausibility [149, 153, 155].

The physics-based simulation category builds on reinforcement learning-based motor control techniques widely used in computer graphics and robotics [85, 86, 128]. These methods reconstruct motion by controlling a virtual character inside a physics simulator, integrating kinematic predictions with physical control to achieve temporally coherent and physically consistent motion [45, 60, 87, 88, 163]. In VR/AR applications, partial observations obtained from HMDs or tracking devices are combined with physics simulation to recover full-body motion in real time [71, 84].

In contrast, physics-integrated learning models incorporate physical laws directly into the inference process. These methods embed rigid-body dynamics, PD control, contact constraints, or dynamics-inspired losses within the neural network, enabling the model to enforce physical consistency during estimation without relying on external simulators or post-processing [72, 122, 123, 167]. Such approaches mitigate issues inherent to image-based pose estimation, such as foot sliding, inconsistent contacts, and unstable center-of-mass behavior by introducing physical knowledge at training.

In summary, physics-based modeling has become a powerful foundation for correcting the physical inconsistencies inherent in learning-based motion estimation, enabling stable full-body reconstruction even from sparse or incomplete observations. Building on this perspective, this thesis explores an approach that integrates kinematic information obtained from a small number of non-vision sensors with physics simulation to achieve dynamically consistent full-body motion estimation.

## Chapter 3

# Wearable Camera-based Pose Estimation

In this chapter, we propose a framework for 3D human pose estimation using a single 360° camera mounted on the user’s wrist. Perceiving a 3D human pose with such a simple setup has remarkable potential for various applications (*e.g.*, daily-living activity monitoring, motion analysis for sports training). However, no existing method has tackled this task due to the difficulty of estimating a human pose from a single camera image in which only a part of the human body is captured, and because of a lack of training data. We propose a method for translating wrist-mounted 360° camera images into 3D human poses. Since we are the first to try this task, we cannot use existing datasets. To address this issue, we use synthetic data to build our own dataset. This solution, however, creates a different problem, that of a domain gap between synthetic data for training and real image data for inference. To resolve this problem, we propose silhouette-based synthetic data generation created for this task. Extensive experiments comparing our method with several baseline methods demonstrated the effectiveness of our silhouette-based pose estimation approach.



Figure 3.1: Concept of our 3D human pose estimation method using a wrist-mounted 360° camera. (a) Future vision depicting a smartwatch with a 360° camera. The area shown in red represents the camera's field of view; (b) Camera view with a 360° camera (GoPro MAX) mounted on the wrist, which is the camera configuration of our method; (c) 3D pose of the user estimated from (b) image by our 3D human pose estimation framework.

### 3.1 Background

Human Pose Estimation (HPE) has long been studied in the computer vision community [20,77]. In particular, 3D pose estimation using a monocular camera, which is one of the most widely used visual sensors in the world, has been actively studied [113] because of its usefulness in various fields such as animation, virtual reality, healthcare, and sports. In the past, many third-person perspective 3D human pose estimation methods have been proposed that use an *outside-in* arrangement of single or multiple cameras statically placed around the user [92,93,102,109]. However, since the third-person perspective method can only estimate the motions within the range of vision of the externally placed camera, it is often not practical in real-world situations where people move around in a large space.

In contrast, there are first-person perspective (also known as “egocentric”) 3D human pose estimation methods [15,16,36,44,46,108,131,150,162], in which cameras or inertial measurement units (IMUs) are attached to the body. These can obtain 3D pose data in various activities in the real world due to their mobility and flexibility, enabling new applications such

as motion recognition and performance analysis in the fields of sports and healthcare. In egocentric human pose estimation, it is important to minimize the number of devices in terms of usability. For this reason, mainstream methods in recent years have been *inside-out* methods [53, 54, 100, 161, 162] in which an outward-facing camera is attached to the body, and *inside-in* methods [44, 48, 108, 131, 132, 140, 150] in which a camera with a fisheye lens is attached to the body to capture the camera wearer’s body with its wide field of view. However, thus far, there is no method with a more practical camera setting, *i.e.*, a single wrist-mounted camera that could be introduced in smartwatches in the future (see Fig. 3.1-(a)).

Therefore, we propose a framework for estimating 3D human poses from images taken with a single wrist-mounted camera. This task is quite challenging as some human body parts are hidden from the camera’s line of sight. We make use of a single 360° camera and a neural network-based framework that leverages time-series features to estimate a 3D human pose with only limited visual information (see Fig. 3.1-(b), 3.1-(c), and Fig. 3.2). The difficulty is how to prepare the training data; there is no existing dataset for 3D human pose estimation with wrist-mounted cameras. Existing works with body-mounted cameras that faced this issue tackled it by using synthetic data [44, 48, 54, 131, 132, 140, 150]. However, training on synthetic datasets, which lack diversity in subject appearance, lighting, or background variation, can inherently cause domain gaps and restrict generalizability [67, 78].

To overcome these issues, we also describe a simplified method for generating training data. In contrast to the conventional approach of creating realistic and diverse large-scale datasets, we propose an approach that bridges the domain gap by using binary silhouette images for both training and inference data. We train the network using silhouette equirectangular image sequences generated using only existing motion capture (MoCap) data. During inference, the 3D poses are estimated from silhou-

ette equirectangular image sequences obtained by our proposed silhouetting process from real images captured by the 360° camera. The proposed method for generating synthetic data is quite simple and thus can be easily applied to any other approach that uses varied camera positions or a different number of cameras.

We validate our approach for egocentric pose estimation on a large motion capture dataset and an in-the-wild dataset consisting of various human motions (walking, crouching, jumping, raising hands, and motion transitions). Experiments comparing the accuracy of pose estimation with several different types of data show the effectiveness of our silhouette-based approach. Experiments comparing the accuracy of pose estimation and the acceleration of the output pose with several different pose estimation models show that our model, which utilizes time-series features, generates smooth and accurate poses.

To summarize, our contributions are as follows:

- We are the first to propose a 3D human pose estimation framework using a single wrist-mounted camera, which contributes to many important applications.
- To bridge the domain gap between synthetic data and real-world data, we describe a method of silhouette-based human pose estimation. Our simplified method of generating silhouette-based training data has the potential to be used for other camera configurations.
- We provide extensive experiments and show the effectiveness of our silhouette-based pose estimation approach and the pose estimation model that leverages time-series features.
- We build a training dataset, composed of 54K frames of silhouette images, for our new problem setting. We make it publicly available to promote progress in the area of egocentric motion capture.

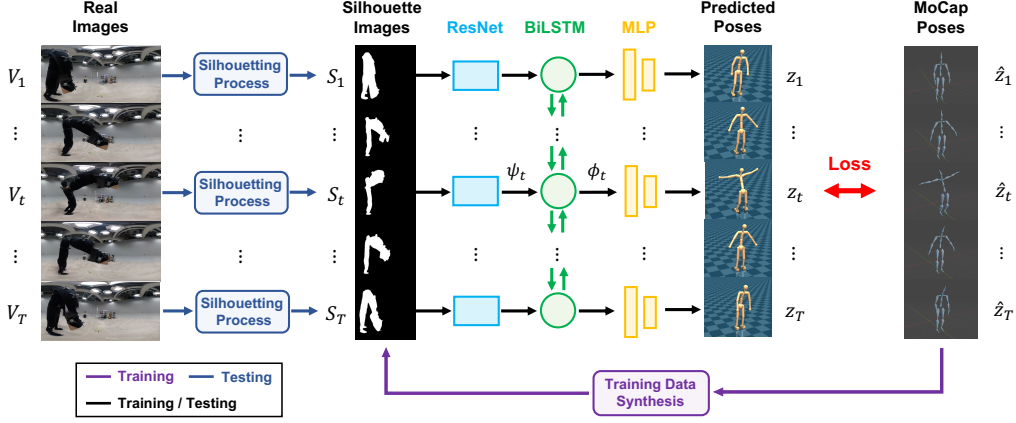


Figure 3.2: Overview of our 3D pose estimation method using a single wrist-mounted 360° camera. The network extracts time-series features from a silhouette image sequence and predicts a 3D human pose sequence. The input data for the network during training is a synthetic silhouette image sequence generated by animating a humanoid using MoCap data as described in Sec. 3.2.2. During inference, the input is a silhouette image sequence obtained by our silhouetting process from equirectangular images captured by a 360° camera in a real environment, as described in Sec. 3.2.3.

## 3.2 Method

As shown in Fig. 3.2, our goal is to estimate 3D human poses  $z_{1:T}$  from a real equirectangular image sequence  $V_{1:T}$  by our network described in Sec. 3.2.1. The network is trained only on synthetic silhouette data generated at a lower cost than conventional methods by the method in Sec. 3.2.2. During inference, the silhouetting process presented in Sec. 3.2.3 is applied to real equirectangular images to bridge the domain gap between synthetic data and real-world data. As an optional feature, we also propose a method for adding noise to the synthetic training data to improve the robustness of the model against noise caused by the silhouetting process and the smoothness of the output pose during inference. Details of this method are in Sec. 3.2.4.

### 3.2.1 Human Pose Estimation Network

Our network  $\mathcal{F}$  takes the input of the silhouette equirectangular image sequence  $S_{1:T}$  and predicts the humanoid state  $z_{1:T}$  at each frame, as shown in Fig. 3.2. The humanoid state  $z_t$  consists of the pose  $p_t$  (position and orientation of the root, and joint angles) and velocity  $v_t$  (linear and angular velocities of the root, and joint velocities). Since human motion is temporally continuous and smooth, the change in poses in nearby past and future frames can be utilized to estimate human poses from images. Therefore, the network extracts image-by-image features from given sequential images and estimates 3D human poses based on their temporal context. This network is based on Yuan and Kitani’s work [162]. The use of time-series features enables smoother and more natural motion estimation than image-by-image pose estimation.

The model encodes the silhouette image by ResNet-18 [40] to extract the feature vector  $\psi_{1:T} \in \mathbb{R}^{128}$  and feeds it to Bidirectional Long-Short Term Memory (BiLSTM) to generate the visual context  $\phi_{1:T} \in \mathbb{R}^{128}$  for each frame. We then feed it to the Multilayer Perceptrons (MLPs) and predict the humanoid state  $z_{1:T}$ . The Mean Squared Error (MSE) is used as the loss function:

$$L(\zeta) = \frac{1}{T} \sum_{t=1}^T \|\mathcal{F}(S_{1:T})_t - \hat{z}_t\|^2, \quad (3.1)$$

where  $\zeta$  is the parameter of this network  $\mathcal{F}$ , and  $\hat{z}_t$  is the ground-truth humanoid state. The optimal  $\mathcal{F}^*$  can be obtained by an SGD-based method.

### 3.2.2 Training Data Synthesis

Fig. 3.3 shows an overview of our process of generating an equirectangular image sequence corresponding to the MoCap data for training the network. In a virtual environment such as one provided by Unity [127], we



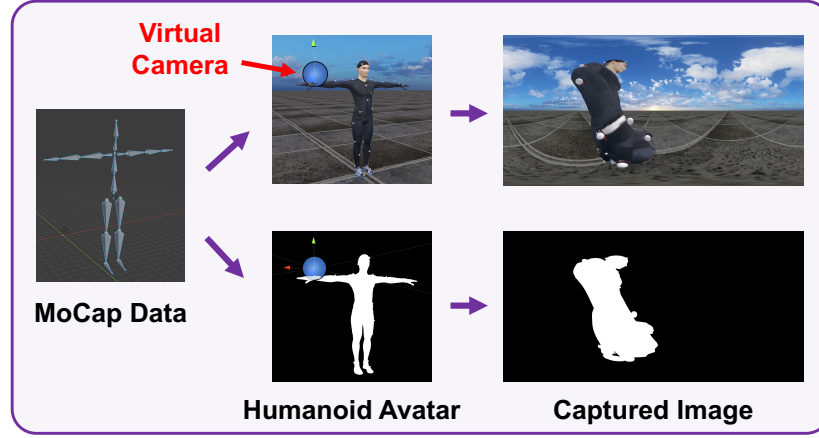


Figure 3.3: Overview of training data synthesis. We animated a humanoid avatar using MoCap data and captured equirectangular images with a virtual  $360^\circ$  camera fixed at the avatar’s wrist position. To capture binary silhouette images, we set the model’s body to white and the background to black.

animate a humanoid avatar by applying the MoCap data. A virtual camera with a  $360^\circ$  field of view is fixed at the wrist of the avatar. Since this virtual camera can be attached to any part of the avatar’s body, our method can be easily adapted to any camera position, not only the wrist. By capturing images while keeping them horizontal, we generate RGB equirectangular images as shown in the upper right of Fig. 3.3.

To improve the performance of generalization on real images during inference, conventional methods generated a large amount of synthetic RGB image data using photorealistic and diverse backgrounds and humanoid textures. In contrast, our approach is to bridge the domain gap by using binary silhouette images for both training and inference data. To this end, we generate silhouetted equirectangular images as shown in the lower right of Fig. 3, by setting the background to black and the avatar’s body texture to white, and capturing images with light shining from all directions. By binarizing those output images and resizing them to the input size of the

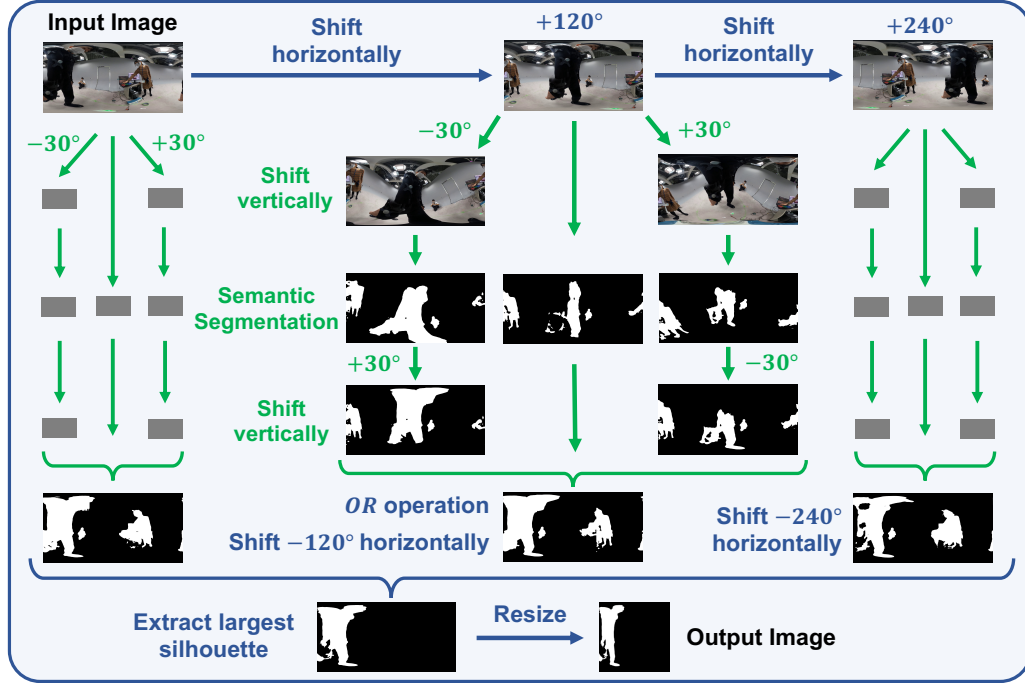


Figure 3.4: Our silhouetting process, which converts an equirectangular image captured by a wrist-mounted  $360^\circ$  camera into a binary silhouette image. We extract the silhouette of the camera wearer from the region labeled as human, which is estimated by an existing semantic segmentation model.

network, we obtain a set of 3D poses and equirectangular image sequences for training.

### 3.2.3 Silhouetting Process for Inference

We propose a silhouetting process for silhouette-based pose estimation, which converts an equirectangular image captured by a wrist-mounted  $360^\circ$  camera into a binary image of a human silhouette. We extract the silhouette of the camera wearer from the region labeled as human, which is estimated by an existing learned semantic segmentation model. In this

study, we used the Swin Transformer [76] model trained on the ADE20K [173] dataset, which is publicly available [94], for semantic segmentation. Since this process only uses the trained model of the semantic segmentation method, it can be replaced by any existing method. The issue here is that images taken with a 360° camera are highly distorted; the top and bottom of the images are stretched out, which is very different from normal-perspective view-images such as ADE20K. Therefore, it is difficult to extract human silhouettes by simply applying semantic segmentation. To solve this issue, we apply semantic segmentation not only to the original equirectangular images but also to the equirectangular images shifted vertically. Then we merge the results of the semantic segmentation to obtain the network input.

Specifically, we first generate three equirectangular images by shifting them horizontally (yaw axis) at 120° intervals. Second, we generate two images by shifting each one by  $\pm 30^\circ$  vertically (pitch axis). Third, we apply semantic segmentation to each image and shift them vertically back to the equirectangular images of the original vertical angle. Fourth, in each of the three horizontally shifted images, we combine the three generated silhouette images via the *OR* operation and shift them horizontally back to the same position as the input image. Finally, we extract the largest silhouette from the three images and resize it for the input of the network.

### 3.2.4 Pseudo-silhouette Generation for Noise Robustness

As an optional feature, we also propose a method of adding generated images of noisy silhouettes (hereafter, “pseudo-silhouettes”) to the training data to improve the robustness of the model against silhouette noise during inference. Although our silhouetting process aims to bridge the domain gap between the synthetic training data and the real data for inference by using silhouettes, it is still difficult to completely bridge the gap. The syn-

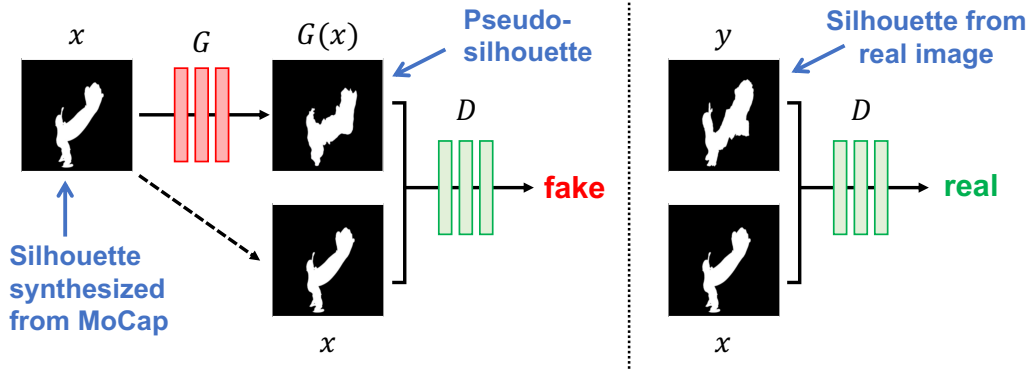


Figure 3.5: Training the pix2pix model to map our synthetic silhouette images with smooth contour to noisy silhouette images obtained through our silhouetting process. The discriminator,  $D$ , learns to classify between fake (synthesized by the generator) and real {smooth silhouette image, noisy silhouette image} tuples. The generator,  $G$ , learns to fool the discriminator.

thetic training data in our method are silhouette images with smooth contours, while the silhouette of the input image during inference has a rough contour (hereafter, “silhouette noise”). The silhouette noise is caused by applying semantic segmentation trained on general perspective view images to equirectangular images and is an inaccurate contour with some missing parts as in the bottom image of Fig. 3.4, or some protruding parts as in the top right image of Fig. 3.5. These differences between the training data and test data affect the accuracy of pose estimation as a domain gap.

Therefore, we improve the robustness of the model by adding pseudo-silhouette images to the training data. The pseudo-silhouette image is generated by converting the smooth silhouette synthetic image to a noisy silhouette image using the pix2pix [52] model, an image-to-image translation method with conditional adversarial networks. The silhouette noise can be reproduced by the pix2pix model trained to transfer the synthetic silhouette image generated using MoCap data to the silhouette image obtained from the real image (see Fig. 3.5). By adding this pseudo-silhouette image

to the training data, even if the image sequence contains silhouette noise during inference, it is less likely to be affected by it, resulting in the estimation of more correct poses and smoother motions.

### 3.3 Experiments

This section presents the experimental settings and evaluation results conducted to validate the effectiveness of the proposed method. We first describe the datasets used, implementation details, and evaluation metrics. We then assess the effectiveness of the proposed approach through two experiments. Experiment 1 evaluates the effectiveness of silhouette-based pose estimation and clarifies the contribution of silhouette information as an input representation. Experiment 2 compares the proposed pose estimation network with existing methods to assess the validity of the designed network architecture.

#### 3.3.1 Dataset

**MoCap Training and Test Data.** We used OptiTrack to capture the motion data to construct the dataset (see Fig. 3.6). Four subjects each wore a 360° camera on their wrist and were asked to perform a variety of actions including walking, jumping, crouching, raising hands, and transitioning between all of these motions. Each take lasted about 5 min and a total of 9 takes were captured, of which 7 takes were used as training data and 2 takes as test data. After removing the parts of inaccurate motions caused by motion capture failures, the training data consists of 54K frames and the test data consists of 14K frames.

**In-the-wild Data.** We also collected in-the-wild data to verify the effectiveness of our method in a real-world environment. As in the previous MoCap

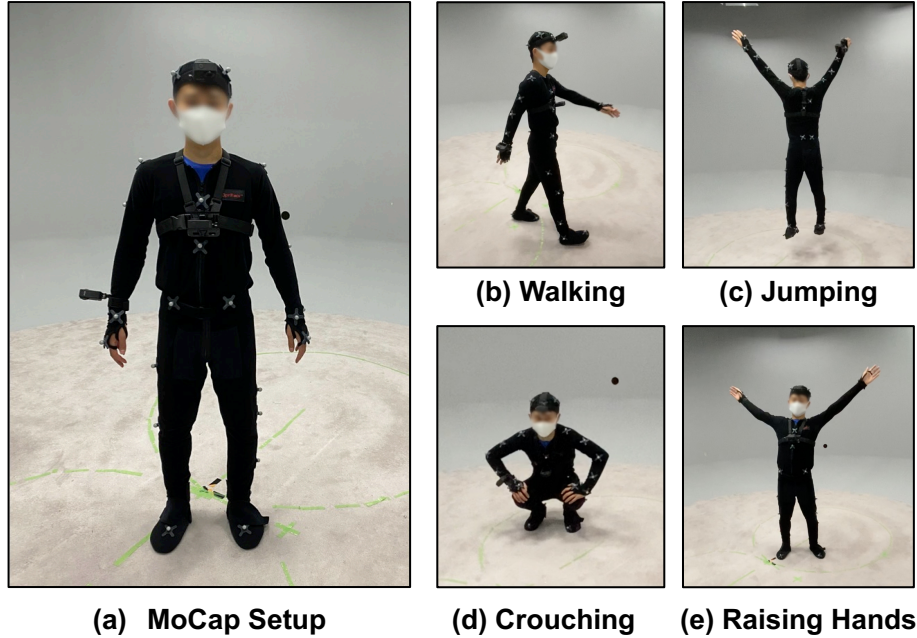


Figure 3.6: Motion capture for constructing our MoCap Training and Test Data. Subjects wore motion capture markers and the 360° camera as shown in Fig. (a) and were asked to perform actions as shown in Fig. (b)-(e).

training and test data collection, the subject wore the 360° camera on the wrist and was asked to perform a variety of actions. This dataset consisted of 11 videos each lasting about 5 sec. As it is hard to obtain ground-truth 3D poses in a real-world environment, following Yuan and Kitani [162], we captured side-view poses of the subject, which were used for quantitative evaluation based on 2D keypoints.

### 3.3.2 Implementation Details

The weights of ResNet-18 [40] used in our 3D human pose regressor were pre-trained with ImageNet [24]. The input equirectangular image was resized to  $224 \times 224$ . The Adam [63] optimizer was employed at the learning

rate of  $1e - 4$ . When training this network, for each time step we sampled data fragments in turn for 120 frames (4 sec) and padded 10 frames of visual features  $\psi_t$  on both sides to reduce the estimation error on the boundary frames when computing  $\phi_t$ . We used Unity for the virtual environment to generate the training data and MuJoCo [130] to visualize estimated human poses that consisted of 52 Degrees of Freedom (DoFs) and 19 rigid bodies.

The pix2pix model for generating the pseudo-silhouettes was trained using our MoCap Training Data. The model was trained to learn a translation from the silhouette image generated by animating the avatar using the MoCap to the silhouette image obtained from the real image by our silhouette process. To align the position of the silhouettes in the real and virtual images, we shifted these images so that the *Sum of Squared Differences (SSD)* was minimized. When training our pose estimation network, we added a pseudo-silhouette image with a probability of one-fourth to the image sequence selected with a probability of one-half at each iteration.

### 3.3.3 Evaluation Metric

To evaluate the accuracy of the pose estimation and the smoothness of the motion of our approach, we use the following evaluation metrics. For the estimated and ground-truth keypoints, we set the hip keypoint as the origin and scaled the coordinate to make the height between the shoulder and hip equal to 0.5 [m].

**Pelvis-aligned Mean Per-Joint Position Error (PEL-MPJPE).** A pose-based metric used by Isogawa *et al.* [51] that measures the Euclidean distance between the estimated 3D pose and the ground-truth 3D pose in mm. We use it for evaluating the accuracy of pose estimation on MoCap test data. This

metric is defined as

$$\text{PEL-MPJPE} = \frac{1}{TJ} \sum_{t=1}^T \sum_{j=1}^J \| (x_t^j - x_t^{\text{root}}) - (\hat{x}_t^j - \hat{x}_t^{\text{root}}) \|_2, \quad (3.2)$$

where  $x_t^j$  is the  $j$ -th joint position of the estimated pose, and  $\hat{x}_t^j$  is the ground truth.  $x_t^{\text{root}}$  and  $\hat{x}_t^{\text{root}}$  represent the root joint position of the estimated and ground-truth poses, respectively.

**2D Keypoint Error ( $E_{\text{key}}$ ).** A pose-based metric proposed by Yuan and Kitani [162] that measures the Euclidean distance between the estimated 2D pose and the ground-truth 2D pose in mm. We use it for evaluating the accuracy of posture estimation on the in-the-wild data. The metric is defined as

$$E_{\text{key}} = \frac{1}{TJ} \sum_{t=1}^T \sum_{j=1}^J \| y_t^j - \hat{y}_t^j \|_2, \quad (3.3)$$

where  $y_t^j$  is the  $j$ -th 2D keypoint of the estimated pose obtained by projecting the 3D joints to an image plane with a side-view camera, and  $\hat{y}_t^j$  is the ground truth extracted with OpenPose [12].

**Acceleration Error ( $E_{\text{accl}}$ ).** A metric proposed by Kanazawa *et al.* [62] that measures the average difference between ground truth and predicted acceleration of each joint in mm/s<sup>2</sup>. For the MoCap test data, we used the original metric  $E_{\text{accl3d}}$  for 3D poses, and for the in-the-wild data, we used the metric  $E_{\text{accl2d}}$  adapted to 2D data. The metric is defined as

$$E_{\text{accl}} = \frac{1}{TJ} \sum_{t=1}^T \sum_{j=1}^J \| a_t^j - \hat{a}_t^j \|_2, \quad (3.4)$$

where  $a_t^j$  and  $\hat{a}_t^j$  are the acceleration of the  $j$ -th keypoint of the estimated pose and ground truth pose, respectively.



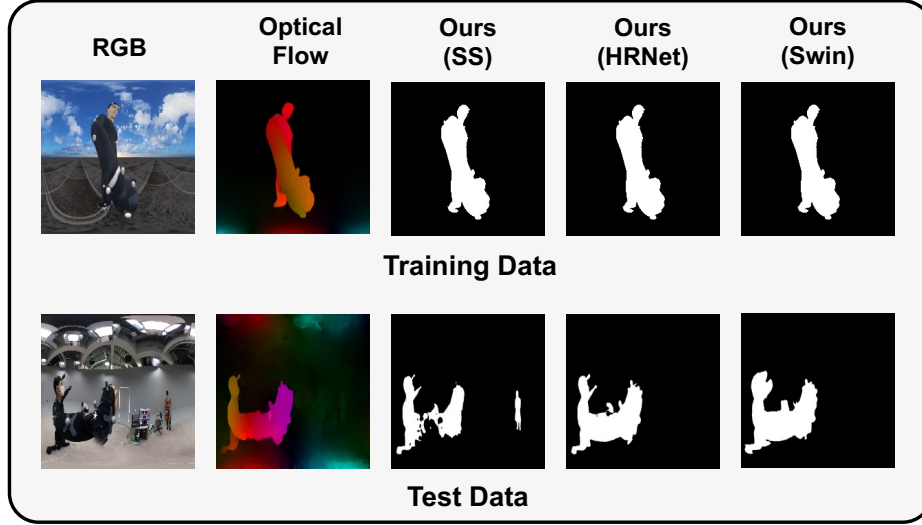


Figure 3.7: Examples of training and test data for each method in experiment 1. We used the same silhouette images for training our silhouette-based methods (SS, HRNet, and Swin).

### 3.3.4 Experiment 1: Investigating the Effectiveness of Our Silhouette-based Pose Estimation

To investigate the effectiveness of our silhouette-based approach, we evaluate the accuracy of the pose estimation when using our network on several different types of data. We compare our method (denoted as **Ours (Swin)** in Table. 3.1) against the following four baseline methods. Fig. 3.7 shows examples of input data used for training and testing, respectively, for each baseline method.

- **RGB:** A method that trains our network on synthetic RGB data and tests it on real RGB data taken with a 360° camera.
- **Optical Flow:** A method that uses optical flow obtained by PWC-Net [125] from image sequences to train our network. This is the same method as PoseReg proposed by Yuan and Kitani [162]. The

optical flows for training and testing were obtained from the image sequences of the RGB method above, respectively.

- **Ours (SS):** A method that trains the network on our synthetic silhouette data and tests it on silhouette images that are human-labeled regions extracted by applying semantic segmentation to real equirectangular images.
- **Ours (HRNet):** A method proposed in [43] that trains the network on our synthetic silhouette data and tests it on silhouette images obtained by our silhouetting process using a semantic segmentation model based on HRNet.

Quantitative results of the pose estimation on the MoCap test data and the in-the-wild data are shown in Table 3.1, and qualitative results are shown in Fig. 3.8. The results show that our method outperforms the baseline methods.

The RGB method had large errors due to the continuous output of incorrect poses such as bending forward throughout the sequence. This occurred because the synthetic training data was not sufficiently photorealistic or diverse with respect to the humanoid avatar’s appearance, background, and lighting, *i.e.*, there was a large domain gap between the test data collected in the real environment.

Although the optical flow method occasionally estimated correct walking and crouching motions, the overall output was quite unstable and inaccurate. This is because the real-world data contains complex optical flows such as objects and lighting around the subject, which bring a domain gap between the training data. In order to bridge this gap, it is necessary to diversify the background objects and lighting in the training data.

The SS method also had lower estimation accuracy than the method using our silhouetting process. Due to the distortion of the equirectangular

Table 3.1: Quantitative results of evaluating the accuracy of pose estimation on the MoCap test data and the in-the-wild data in Experiment 1. Bold values show the best scores.

| MoCap Test Data (PEL-MPJPE)    |             |              |              |             |                                    |
|--------------------------------|-------------|--------------|--------------|-------------|------------------------------------|
| Method                         | Walk        | Crouch       | Jump         | Raise Hands | All Frames                         |
| RGB                            | 209.7       | 209.7        | 215.5        | 258.6       | $212.8 \pm 69.5$                   |
| Optical Flow                   | 200.7       | 166.4        | 222.0        | 197.4       | $197.1 \pm 77.1$                   |
| Ours (SS)                      | 141.6       | 142.5        | 181.4        | 128.1       | $144.0 \pm 67.2$                   |
| Ours (HRNet)                   | 99.0        | 160.2        | 163.9        | 110.4       | $113.3 \pm 61.5$                   |
| <b>Ours (Swin)</b>             | <b>83.1</b> | <b>104.7</b> | <b>116.1</b> | <b>95.5</b> | <b><math>89.4 \pm 34.9</math></b>  |
| In-the-wild Data ( $E_{key}$ ) |             |              |              |             |                                    |
| Method                         | Walk        | Crouch       | Jump         | Raise Hands | All Frames                         |
| RGB                            | 330.1       | 263.1        | 245.1        | 175.8       | $225.6 \pm 82.3$                   |
| Optical Flow                   | 359.3       | 306.0        | 320.1        | 361.4       | $337.4 \pm 67.3$                   |
| Ours (SS)                      | 119.0       | 134.5        | 149.2        | 109.8       | $130.4 \pm 61.1$                   |
| Ours (HRNet)                   | <b>88.2</b> | 125.6        | 121.5        | 103.1       | $110.4 \pm 42.5$                   |
| <b>Ours (Swin)</b>             | 88.3        | <b>119.3</b> | <b>117.0</b> | <b>95.0</b> | <b><math>106.1 \pm 51.3</math></b> |

images, the segmentation of human regions often failed, resulting in very noisy data with undesired regions other than the camera wearer’s body.

In contrast, our proposed methods using silhouetting processes (HRNet and Swin) were able to generate 3D human poses close to the ground truth. In particular, using the Swin Transformer model, *i.e.*, a state-of-the-art semantic segmentation method, we were able to correctly extract the silhouette and estimate the pose with high accuracy.

These results show that our silhouette-based pose estimation method using synthetic training data and a silhouetting process for inference works well in bridging the domain gap between synthetic data and real data.

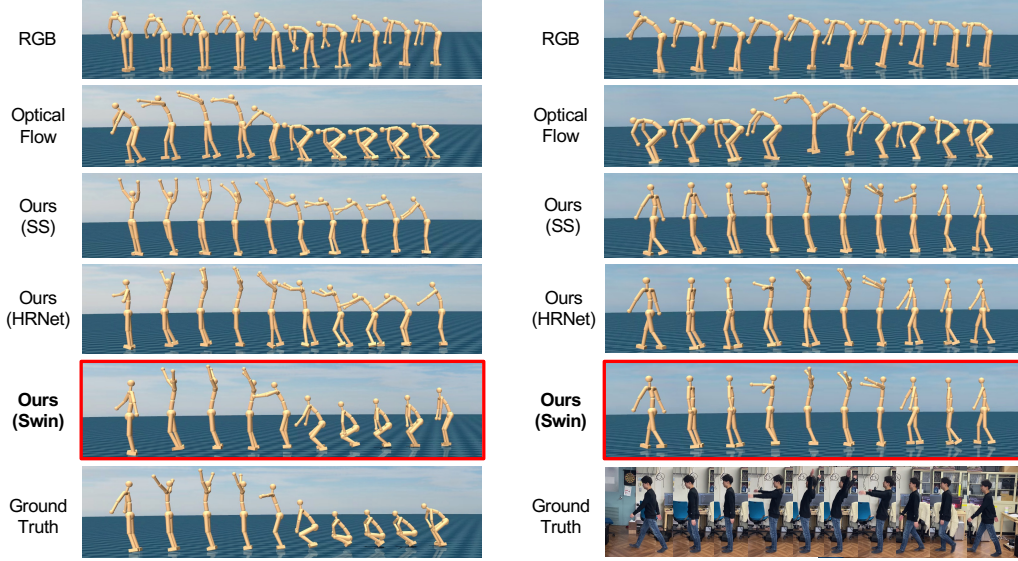


Figure 3.8: Qualitative results for each method on the MoCap test data (left) and the in-the-wild data (right) in Experiment 1.

### 3.3.5 Experiment 2: Investigating the Effectiveness of Our Pose Estimation Network

To verify the effectiveness of our pose estimation model that uses time-series features, we conduct an experiment to evaluate the accuracy of the pose estimation and the smoothness of the output pose. However, there is currently no 3D human pose estimation method using a wrist-mounted 360° camera, and there is no publicly available code for other egocentric 3D human pose estimation methods applicable to this experiment. Therefore, as baseline methods, we used third-person-view-based existing 3D human pose estimation networks. Please note that we re-trained those networks with the same synthetic silhouette image dataset as our method for a fair comparison. We evaluate the following four baseline methods, including our model trained with pseudo-silhouette images in the training data.

- **Mobile Human Pose:** A model proposed by Choi *et al.* [21] for real-time 3D human pose estimation from a single image, which is the most precise and compact model that can be implemented in mobile devices.
- **Integral Human Pose:** A model proposed by Sun *et al.* [126] for 3D human pose estimation from a single image, which is a simple and effective integral regression model that unifies the heatmap representation and the joint regression approach, sharing the merits of both.
- **Ours w/o Pseudo-Silhouette (PS):** Our proposed model, which was trained using only smooth synthetic silhouette images without pseudo-silhouette images.
- **Ours w/ Pseudo-Silhouette (PS):** Our proposed model aims to be robust against silhouette noise by mixing pseudo-silhouette images in the training data.

For Experiment 2, quantitative results are shown in Table 3.2 and qualitative results are shown in Fig. 3.9. From the results in Table 3.2, the highest accuracy of pose estimation was achieved by the Integral method for the MoCap Test Data and our method (Ours w/ PS) for the In-the-wild Data. In contrast to the Integral method, which estimates joint positions, our method, which estimates joint angles, can cause accumulation of joint angle estimation errors in end joints such as hands and feet. While this may be the reason for the inferior accuracy of our method, the differences between them were not very large.

On the other hand, in the comparison of acceleration, our proposed method is much smoother than the Mobile and Integral methods. This is because the Mobile and Integral methods estimate the pose frame-by-frame, and the accuracy of the pose estimation is significantly decreased for images with severe noise or occlusion. In contrast, our network utilizes

Table 3.2: Quantitative evaluation results of pose estimation accuracy and joint acceleration on the MoCap test data and the in-the-wild data in Experiment 2.

| Method             | MoCap Test Data                   |                     | In-the-wild Data                   |                     |
|--------------------|-----------------------------------|---------------------|------------------------------------|---------------------|
|                    | PEL-MPJPE                         | $E_{\text{accl3d}}$ | $E_{\text{key}}$                   | $E_{\text{accl2d}}$ |
| Mobile             | 99.7 $\pm$ 48.3                   | 55.8                | 120.8 $\pm$ 58.2                   | 85.0                |
| Integral           | <b>83.2 <math>\pm</math> 44.8</b> | 46.5                | 107.5 $\pm$ 49.2                   | 78.5                |
| <b>Ours w/o PS</b> | 89.4 $\pm$ 34.9                   | 24.3                | <b>106.1 <math>\pm</math> 38.5</b> | 51.3                |
| <b>Ours w/ PS</b>  | 86.2 $\pm$ 35.4                   | <b>17.5</b>         | <b>106.1 <math>\pm</math> 47.6</b> | <b>48.1</b>         |

time-series features for inference, which results in smooth and stable motion estimation even when such images are included in the sequence. In addition, our method (Ours w/ PS) has the lowest error among the  $E_{\text{accl}}$ , indicating that the smoothness of the output poses is improved by mixing pseudo-silhouette images during training. These results demonstrate the effectiveness of our proposed method, *i.e.*, a pose estimation network that takes advantage of time series features and pseudo-silhouette images to improve the robustness of the model against silhouette noise.

### 3.4 Discussion and Limitations

We proposed a new 3D pose estimation method using only a single 360° camera attached to the wrist for various applications such as motion recognition and motion analysis in the fields of sports, medicine, animation, and others. However, in order to use it for applications in real-world scenarios, it will be necessary to resolve the following issues.

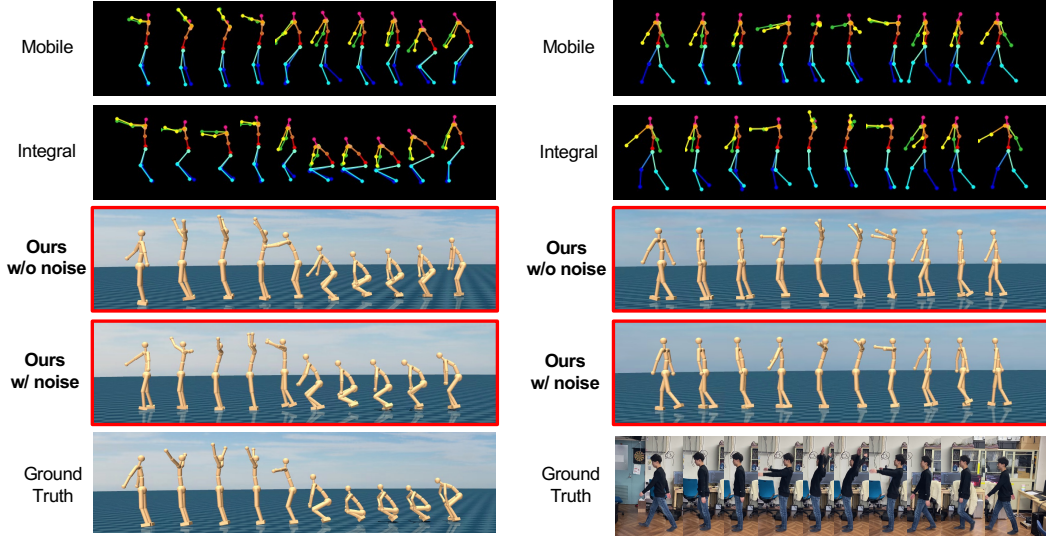


Figure 3.9: Qualitative results for each method on the MoCap test data (left) and the in-the-wild data (right) in Experiment 2.

### 3.4.1 Practical Size and Weight of Sensor Devices

Although we proposed an approach that utilizes a minimal number of devices, the commercial 360° camera we used in our experiment was not small and light enough to be usable on a daily basis on a wrist. However, since there are already smartwatches equipped with cameras, we believe that we may soon see a wristwatch-type device equipped with a 360° camera that can capture the wearer’s body with a wide field of view, as shown in Fig. 3.1-(a).

### 3.4.2 Versatile and Highly Accurate Pose Estimation

Although it is important to estimate various human poses for practical use, our model is currently not capable of estimating all human motions. However, since our method uses only MoCap data to generate synthetic training

data, it is quite easy to prepare a more diverse data set using publicly available MoCap data. We will verify how well our method is able to estimate complex or unusual motions in a future study.

As for accuracy, there is a limit to silhouette extraction accuracy because we use a semantic segmentation model trained on existing third-person images for egocentric equirectangular images. However, by leveraging the fact that deep-learning-based foreground extraction is robust to domain shift [104], it is possible to create a model that extracts silhouettes with high accuracy using synthetic data.

In addition, fast arm swings or low-light environments may cause motion blur, leading to failures in silhouette extraction. This issue is inherent to frame-based cameras, as they rely on shutter-based image capture. A promising solution is to employ sensing modalities based on different principles rather than conventional frame-based imaging.

It is also important to estimate the user's global position. Several camera localization methods have been proposed [16, 36, 140], and they could potentially be integrated into our method. However, this requires addressing the challenge that, in our setup, the camera undergoes more complex motion due to its wrist-mounted placement, unlike conventional head- or chest-mounted configurations.

### 3.4.3 Real-Time Pose Estimation

Real-time pose estimation technology enables a wide range of new applications. While our method can estimate poses from silhouette images in real time (approximately 100 FPS on an Intel Core i9 and a GeForce RTX 3090), the silhouette generation process requires substantial computation for person region extraction involving multiple geometric transformations, and thus runs at only about 0.1 FPS, becoming a major bottleneck in the overall



system. To address this issue, adopting approaches such as that of Wang *et al.* [139], which partition distorted fisheye images into locally undistorted patches and leverage vision transformers to efficiently estimate full-body and hand skeletons as well as human mesh models, is expected to significantly improve inference speed.

### 3.5 Summary

We presented a framework for estimating 3D human poses using a single wrist-mounted 360° camera. Our pose estimation network is trained only on synthetic silhouette image data generated in the virtual environment. For inference, our method uses binary silhouette images generated via a silhouetting process that takes real equirectangular images as input. Our silhouette-based method could reduce data generation costs and bridges the domain gap between synthetic and real data, which has been an issue in previous research. As an optional feature, we also proposed pseudo-silhouette image generation for noise robustness using conditional adversarial networks. Although we are currently assuming that there are real images synchronized with the MoCap data, we plan to train a generalized model by generating a larger dataset in the future. The experimental results have demonstrated the effectiveness of our silhouette-based approach and our pose estimation network, which takes advantage of the time-series features of the image sequences.

While our method provides a practical solution for egocentric pose estimation with a wrist-mounted camera, its reliance on frame-based imaging leaves it sensitive to motion blur and challenging lighting conditions. To address this limitation, the next chapter explores an event-based approach that operates on fundamentally different principles and offers improved robustness in such environments.

# Chapter 4

## Event-based Pose Estimation

How much can we infer about human shape using an event camera that only detects the pixel position where the luminance changed and its timestamp? This neuromorphic vision technology captures changes in pixel values at ultra-high speeds, regardless of the variations in environmental lighting brightness. Existing methods for human mesh recovery (HMR) from event data need to utilize intensity images captured with a generic frame-based camera, rendering them vulnerable to low-light conditions, energy/memory constraints, and privacy issues. In contrast, we explore the potential of solely utilizing event data to alleviate these issues and ascertain whether it offers adequate cues for HMR. This is a quite challenging task due to the substantially limited information ensuing from the absence of intensity images. To this end, we propose EventPointMesh, a framework which treats event data as a three-dimensional (3D) spatio-temporal point cloud for reconstructing the human mesh. By employing a coarse-to-fine pose feature extraction strategy, we extract both global features and local features. The local features are derived by processing the spatio-temporally dispersed event points into groups associated with individual body segments. This combination of global and local features allows the framework to achieve a more accurate HMR, capturing subtle differences

in human movements. Experiments demonstrate that our method with only sparse event data outperforms baseline methods.

## 4.1 Background

With the rise of virtual reality (VR), augmented reality (AR), metaverse applications, and other immersive experiences like sports viewing and stage performances, understanding a human pose and shape non-invasively has become paramount. These immersive applications not only enhance our digital social interactions but also revolutionize content creation, animation, training, and rehabilitation. However, the challenge is to capture human states, including 3D pose and shape, accurately across various real-world settings to elevate the depth of the immersion in these digital realms.

So far, optical-based motion capture (MoCap) systems [82,99] have offered high-precision and high-speed capturing capabilities. However, their primary challenge remained the associated high costs. With the progression of technology, there has been a notable drift towards data-driven pose estimation methods [11,27,141], with techniques leveraging affordable RGB cameras gaining traction. These pose estimation techniques predominantly work with sparse 3D joint representations in the form of skeletal models. While these skeletal models can represent simple actions relatively clearly, they fall short when detailing intricate human behaviors. The need to describe the human body with finer granularity has led to a renewed focus on HMR methods [62,65,129]. HMR aims to delve deeper, estimating intricate details of human form, including the 3D pose and shape. However, a significant challenge with RGB-based methods remains their heavy dependence on visible light, which leads to issues such as object occlusions, lighting condition constraints, privacy issues in some scenarios, and considerable power and memory demands hindering edge device deployment. As a solution, researchers have been exploring HMR techniques

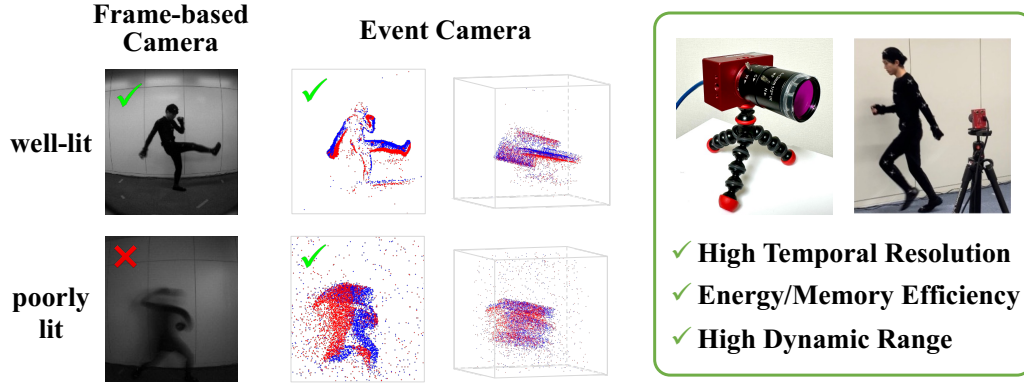


Figure 4.1: Event cameras are designed to detect luminance changes, offering high temporal resolution and a high dynamic range. These unique features effectively address the challenges faced by frame-based cameras, such as motion blur and low frame rates in poorly-lit environments.

using alternative modalities such as wireless signals. Notably, methods leveraging radio frequency (RF) signals, including Wi-Fi and millimeter-wave (mmWave) [151, 170], have gained attention. These methods take advantage of longer wavelengths than visible light, reducing memory and personal data consumption. However, they pose challenges, including potential restrictions in environments with sensitive electronic equipment, such as hospital rooms or aircraft.

The use of event-based cameras, henceforth event cameras, has the potential to be a solution to these challenges. Unlike conventional cameras, event cameras detect changes in scene brightness, generating event data including coordinates, time, and polarity. They independently monitor luminance changes for each pixel, recording only significant changes asynchronously. This mechanism ensures high temporal resolution and low power consumption. Furthermore, focusing solely on brightness changes, event cameras have a high dynamic range, making them resilient in low-light conditions, as shown in Fig. 4.1. From these characteristics, event cameras have been incorporated into various research areas, including track-

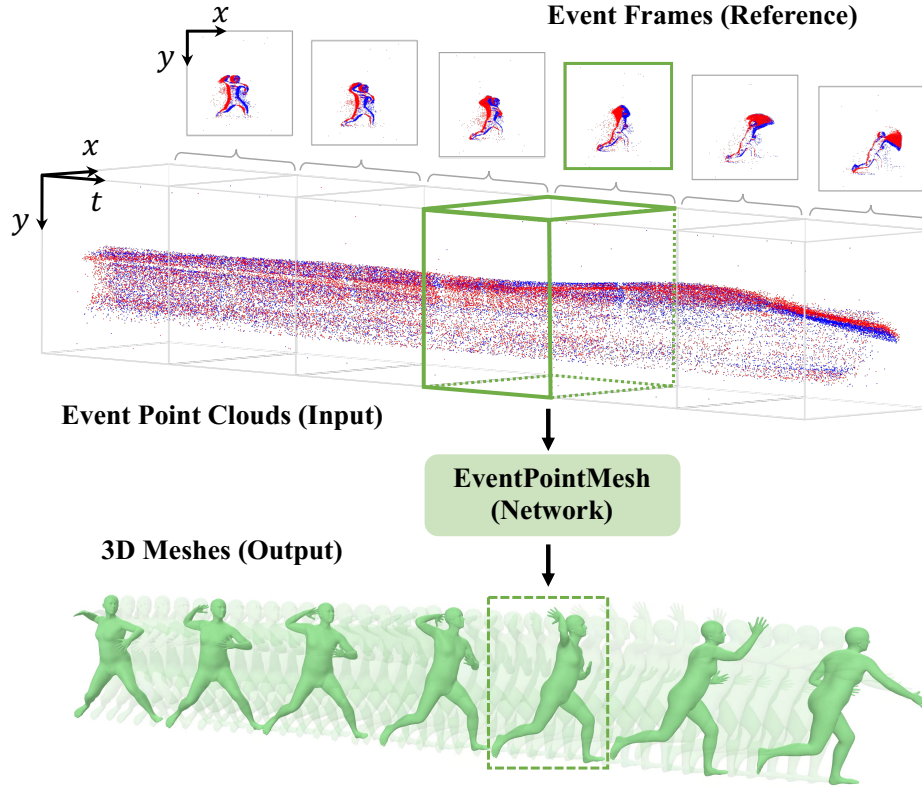


Figure 4.2: We propose EventPointMesh, a method for 3D human mesh recovery solely from event data. EventPointMesh treats event data as point clouds and estimates human meshes from these point clouds segmented by fixed-time intervals. It enables high-speed HMR that is unaffected by lighting conditions.

ing, recognition, 3D reconstruction, robotics, VR/AR, and autonomous driving [31]. Our focus is on employing event cameras for HMR. While there have been proposals for HMR using event data [148,175], these methods utilize grayscale frame images captured simultaneously with event data for initial pose and shape estimation at each time step. This limits the anticipated resilience to dark conditions and energy/memory efficiency of event-based methods.

In this chapter, we tackle the challenge of HMR using only event data,

aiming to develop an HMR method that can operate under various lighting conditions, consumes less power and memory, and also preserves privacy. As illustrated in Fig. 4.2, event data consists of points distributed over the three-dimensional spatiotemporal domain defined by the image-plane coordinates  $(x, y)$  and the temporal axis  $t$ . Each point is associated with a polarity value indicating the direction of luminance change, where red denotes increased luminance and blue denotes decreased luminance. In previous event-based HMR methods [175], event streams were aggregated into fixed temporal intervals and treated as “event frames,” which could potentially discard important motion cues inherent in the high temporal resolution of event data. This aggregation introduces a structural trade-off between temporal resolution and information density, as accumulating a sufficient number of events for reliable spatial representation inevitably compresses temporal information, effectively reducing the achievable temporal resolution.

To overcome this limitation, we propose EventPointMesh, a new framework that treats event data as sparse point clouds, hereafter referred to as “event point clouds”, and estimates human meshes directly from them. Event point clouds differ fundamentally in structure from frame-based images, which makes it impossible to directly apply conventional machine learning models designed for image frames. To address this issue, EventPointMesh adopts PointNet [17], a model capable of directly processing 3D point clouds, as its backbone to handle event data as spatiotemporal point clouds. However, event cameras typically generate hundreds of thousands of events per second, far exceeding the point counts commonly handled by PointNet, which makes processing the entire event stream at once impractical due to GPU memory constraints. To overcome this limitation, our framework divides the event stream into short, fixed-length time windows. For each window, we extract the corresponding spatiotemporal event point cloud and downsample it to remain below a specified point count. The re-

sulting point cloud is then used to estimate the SMPL pose corresponding to that time window.

Within the network, a two-stage feature extraction process is conducted on these point cloud blocks to estimate human meshes, traversing through multiple modules. The first stage involves extracting coarse global features from the point clouds, which contain information about the human’s location, body shape, and approximate pose. The next stage involves a mechanism for extracting fine point cloud features to precisely replicate the human pose. Initially, a module within the network estimates 2D joint positions on the image plane. Then, for each joint individually, another module groups the event points around that specific joint and extracts local features for each of these groups. By combining the global and local features obtained separately, the method ultimately estimates the human’s accurate pose, body shape, and location in 3D space. This coarse-to-fine feature extraction approach enables the retrieval of useful information solely from point clouds, allowing this method to faithfully depict human dynamics from sparse event point clouds without the need for intensity images. Additionally, the event point clouds input to the EventPointMesh network can be set at any temporal width, harmonizing with the high temporal resolution characteristic of event cameras. This facilitates the realization of the HMR method capable of restoring high-frequency 3D poses and shapes of diverse movements in poor lighting conditions.

Furthermore, we propose EventPointMesh Dataset (EPMD), a large-scale HMR dataset comprising event data, intensity images, optical MoCap data, and human mesh models. While there have been event-based HMR datasets before, what sets EPMD apart is its capturing of diverse motion and shape data in both well-lit and poorly-lit conditions. This is the first dataset of its kind to incorporate such lighting variations. Extensive experimental evaluations using the existing large-scale event-based HMR dataset, Multi-Modality Human Pose and Shape Dataset (MMHPSD) [175],

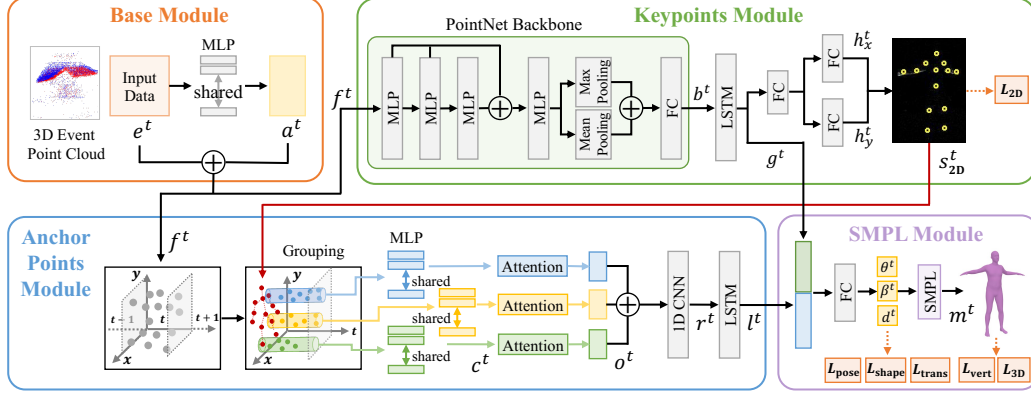


Figure 4.3: Pipeline of our event-based HMR method, EventPointMesh. It consists of four modules: the Base Module, Keypoints Module, Anchor Points, and SMPL Module (Sec. 4.2.1). To achieve highly accurate mesh reconstruction, EventPointMesh employs a coarse-to-fine feature extraction process: the Keypoints Module extracts global features from the event point cloud, and the AnchorPointsModule extracts local features.

and our EPMD, under both well-lit and poorly-lit conditions, have validated the effectiveness of our pose and shape estimation method solely using event point clouds.

In summary, our contributions are as follows: (1) We are the first to tackle the challenge of 3D HMR using only event data. (2) To this end, we propose EventPointMesh, a framework that treats event data as 3D spatio-temporal point clouds for mapping it to human meshes. (3) To faithfully restore human motion, we explicitly extract global and local features from event point clouds, utilizing 2D joint positions as cues for grouping. (4) Given the absence of prior methods for executing this task, we have created EPMD, a dataset for HMR comprising a vast amount of event data, intensity images, optical MoCap data, and mesh data captured under both well-lit and poorly-lit conditions; and (5) We conduct extensive experimentation and show the effectiveness of our method.



## 4.2 Method

We present EventPointMesh, a framework for estimating 3D human meshes from the spatiotemporal point clouds generated by an event camera. As shown in Fig. 4.3, given a sequence of event point clouds  $e^{1:T}$  segmented into  $T$  fixed-width time windows, EventPointMesh estimates the SMPL mesh sequence  $m^{1:T}$ . The network is organized into four modules: the Base Module that extracts point cloud features  $f^t$  from the event point cloud  $e^t$  within the time window indexed by  $t$ ; the Keypoints Module which derives a global feature vector  $g^t$  and predicts 2D human joint positions  $s_{2D}^t$  from  $f^t$ ; the Anchor Points Module designed to group the point cloud around each estimated 2D human joint positions  $s_{2D}^t$  and subsequently extract local features  $l^t$ ; and the SMPL Module that infers the SMPL mesh model from the assembled global and local feature vectors  $g^t$  and  $l^t$ .

EventPointMesh adopts an approach to extract pose features from point clouds in a coarse-to-fine manner. The global features  $g^t$ , extracted in the initial phase through the Keypoints Module, represent coarse features such as the subject’s location, body shape, and approximate pose derived from the event point clouds. These features are utilized for mesh estimation in the subsequent SMPL Module. The 2D joint positions  $s_{2D}^t$ , estimated by the Keypoints Module, are used in the subsequent Anchor Points Module to accurately replicate the human pose. Specifically, based on the 2D joint positions  $s_{2D}^t$ , the event point clouds surrounding each joint are grouped, and local features  $l^t$  are extracted from each group. This process extracts fine features from events triggered by the movement of each body part, which are also utilized in the subsequent SMPL Module. Through such a coarse-to-fine feature extraction process, EventPointMesh is capable of replicating detailed human mesh reconstructions of diverse human movements.

Further elaboration on each module will be provided in the following subsections. In addition, the final subsection introduces our dataset

EPMD, encompassing event data recorded under diverse lighting conditions, intensity images, optical MoCap data, and the SMPL mesh models.

### 4.2.1 Module Details

**Base Module.** This module serves as a component for converting the input event point clouds into high-dimensional features. Each time a luminance change is detected, the event camera generates 4D event data, which includes 2D pixel coordinates, detected timestamp, and polarity indicating the direction of the luminance change. Depending on the specifications of the event camera, the timestamp of luminance change detection is asynchronously recorded for each pixel.

Given the event point cloud segmented into fixed time windows, represented as  $e^t$ , the module transforms it into high-dimensional features using a Multi-Layer Perceptron (MLP). Considering the  $n$ -th detected point within the point cloud as  $e_n^t$ , this point is transformed into a high-dimensional feature vector  $a_n^t = \text{MLP}(e_n^t; w_a)$  by applying the MLP transformation with parameters  $w_a$ . This feature vector is then concatenated with the 4D vector of  $e_n^t$ , resulting in a high-dimensional vector  $f_n^t$ , which is then fed into the subsequent modules. This design aims to enhance the expressiveness of the feature vectors used in subsequent processing by combining the spatiotemporal relationship of point clouds acquired during the learning process with the original attributes of each point (image coordinates, timestamp, polarity). Additionally, the preservation of the original point cloud data, along with the feature vectors, serves another purpose: the subsequent Anchor Points Module requires the spatial information of point clouds.

**Keypoints Module.** For accurately estimating human poses from event data, it is crucial to have a clue as to which event point corresponds to

the movement of a specific body part. Hence, the Keypoints Module estimates 2D coordinates of major joints from the input point clouds. These estimated coordinates serve as reference points (hereafter referred to as “anchor points”) in subsequent modules, aiding in obtaining local features of the point clouds generated by body movements around each joint. This module is designed based on the network proposed by Chen *et al.* [18], considering the temporal features and the overall point cloud feature extraction.

Initially, the feature vector  $f_n^t$  of each event point cloud obtained by the Base Module is transformed into a global feature of the point cloud,  $b_n^t = \text{PointNet}(f_n^t)$  with a network that has PointNet [17] as its backbone. Afterward, a feature vector considering temporal dynamics of the event point cloud sequence is extracted from  $b^t$  via a Bidirectional Long Short-Term Memory (BiLSTM). One of the challenges in working with event cameras is their inherent limitation: they capture only moving body parts, ignoring the stationary ones. This results in sparse data and complicates pose estimation due to the absence of complete body part information. The method by Chen *et al.* [18] faces challenges due to its estimation of 2D joint positions for only one frame per time window. To address this issue, we introduced a BiLSTM to construct a model that is robust against the absence of events from stationary parts, taking into account temporal information. Our method leverages features of point clouds from both forward and backward time windows, effectively compensating for these sparse areas. With the weight parameters of the BiLSTM  $w_g$ , the temporal dynamics aware feature vector can be represented as  $g^{1:T} = \text{BiLSTM}(b^{1:T}; w_g)$ , where  $T$  is the number of time windows used as input. While Chen *et al.* [18] utilize this feature vector for 2D joint position estimation, we view it as a spatiotemporal feature of 3D point clouds that encapsulates information about a person’s location, along with their approximate pose and body shape. Consequently, we employ it as a global feature vector for the

estimation of 3D meshes in the subsequent SMPL Module.

Subsequently, the time-series feature vector  $\mathbf{g}^t$  is transformed into two 1D heat-vectors,  $(\mathbf{h}_x^t, \mathbf{h}_y^t) = \text{FC}(\mathbf{g}^t; \mathbf{w}_h)$ , using fully connected layers (FC) with parameters  $\mathbf{w}_h$ . These heat-vectors represent the confidence distribution of joint positions along the  $x$  and  $y$  axes in an image plane. Chen *et al.* [18] introduced this representation based on the novel coordinate representation, SimCC [74]. Unlike conventional methods that estimate joint positions using 2D heatmaps, this representation allows for highly accurate estimation even in low-resolution images. The heat-vectors are denoted as  $\mathbf{h}_x^t = [h_0, h_1, \dots, h_i, \dots, h_W]$  and  $\mathbf{h}_y^t = [h_0, h_1, \dots, h_j, \dots, h_H]$ , where  $i$  and  $j$  are the indices of the  $x$ -axis and  $y$ -axis in the image of width  $W$  and height  $H$ , respectively. The 2D joint positions  $\mathbf{s}_{2D}^t = (\mathbf{s}_x^t, \mathbf{s}_y^t)$  are then determined by finding the coordinates with the maximum value in each heat-vector. Specifically, the coordinates are obtained by  $\mathbf{s}_x^t = \arg \max_i(\mathbf{h}_x^t(i))$  and  $\mathbf{s}_y^t = \arg \max_j(\mathbf{h}_y^t(j))$ .

**Anchor Points Module.** In order to capture the detailed features of the point cloud and to achieve accurate mesh estimation, we utilize anchor points, which are the 2D joint coordinates estimated by the Keypoints Module. These points facilitate the grouping of event point clouds scattered across the 3D spatiotemporal space. This approach enables us to classify events stemming from body movements around each joint and extract localized features. This network configuration is inspired by mmMesh, proposed by Xue *et al.* [151], which estimates 3D human meshes from millimeter-wave point clouds. Whereas mmMesh utilized fixed-size 3D grid-based anchor points within an  $xyz$  spatial framework, our approach adopts 2D joint points as anchor points, incorporating the temporal dimension into the  $xy$  image plane. These 2D joint positions are then used to group surrounding event points in a 3D spatiotemporal space, based on their Euclidean distance in 2D image coordinates. The details of this process are

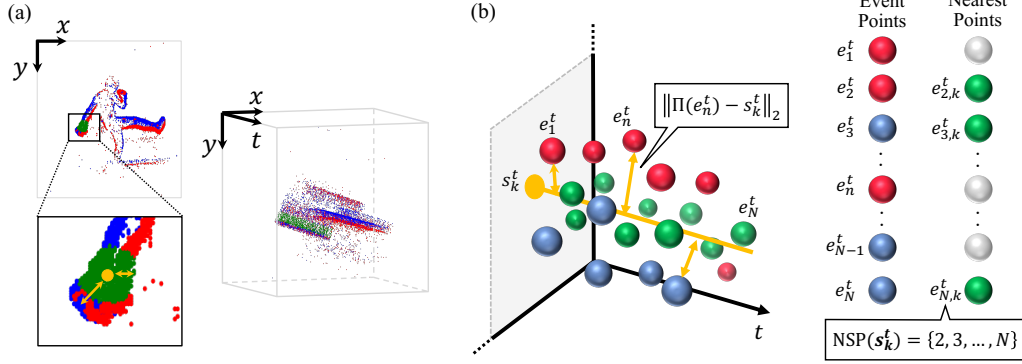


Figure 4.4: Detailed illustrations of the spatiotemporal grouping of event point clouds in the Anchor Points Module. Yellow points represent the 2D joint positions estimated by the Keypoints Module, while red and blue indicate the input event point clouds, and green represents the grouped points. (a) An example of applying spatiotemporal grouping to real event data, focusing on the right wrist, though in practice, this process is applied individually to all major joints. (b) A detailed view of the nearest points selection. For each point in the event point clouds scattered in space and time, the L2 norm on the image plane with the 2D joint coordinates is calculated, and the nearest  $G$  points with the closest distances are selected as a group.

illustrated in Fig. 4.4. For clarity in notation: the coordinates of the  $k$ -th anchor point at the time window  $t$  are denoted as  $s_k^t \in \mathbb{R}^2$ . The set of indices of the grouped nearest  $G$  points is denoted as  $\text{NSP}(s_k^t)$ . A grouped event point is referred to as  $e_{n,k}^t$  where  $n \in \text{NSP}(s_k^t)$ , and its 2D coordinates on the image plane are given by a function  $\Pi$  as  $\Pi(e_{n,k}^t) \in \mathbb{R}^2$ . Using these definitions, the feature vector  $c_{n,k}^t$  is derived through an MLP operation as  $c_{n,k}^t = \text{MLP}([s_k^t, \|\Pi(e_{n,k}^t) - s_k^t\|_2, f_n^t]; w_c)$ . The input to the MLP is a vector formed by concatenating the anchor point coordinates, the distance between the grouped points and the anchor point, and the feature vector from the Base Module. This design enhances our model's capability to recognize the spatiotemporal relationships between anchor points and their neighboring points.

The feature vector  $c_n^t$  is then fed into the attention network to extract particularly important event point features from the grouped point cloud. In existing point cloud based approaches, such as PointNet [17], the Max Pooling operation is typically used to extract the most prominent features by eliminating redundant information. However, as the point cloud grouped by this module already contains less redundant information, we have adopted an attention mechanism to aggregate the features of all points in the group while preventing the loss of important information due to the Max Pooling operation. Representing the attention network as a mapping function  $\text{ATTN}$  with weight parameters  $w_o$ , the feature vector  $c_n^t$  of each grouped point is transformed into the following feature vector:

$$o_k^t = \sum_{n \in \text{NSP}(s_k^t)} \text{ATTN}(c_n^t; w_o) \cdot c_n^t.$$

Furthermore, to aggregate the feature vectors  $o_k^t$  of each anchor point and transform them into features considering the spatial relationship between each group, i.e., the events caused by each body joint, we introduce a 1D convolution layer and obtain the feature vector  $r^t = \text{1DCNN}(o^t; w_r)$ . This  $r^t$  is transformed into a temporal relationship aware feature vector by using BiLSTM, similar to the Keypoints Module. Letting  $w_l$  be the weight parameters of BiLSTM, the module obtains final local feature vector  $l^{1:T} = \text{BiLSTM}(r^{1:T}; w_l)$ .

**SMPL Module.** The primary role of the SMPL Module is to estimate the SMPL mesh model [79] based on the global and local features of the event point clouds obtained from the modules described above. By feeding these features into an MLP, we derive vectors representing a human pose, shape, and translation. Subsequently, by using the SMPL regression model [79], we estimate mesh vertices and 3D human joint positions. The SMPL model has gained traction in contemporary research on pose and shape estimation. Its strength lies in its ability to represent the complex and diverse

poses and shapes of the human body using a limited number of parameters. The model parameters encompass pose parameters  $\theta \in \mathbb{R}^{24 \times 3}$ , indicating the relative rotations of 23 joints and the global rotation of the root joint, and shape parameters  $\beta \in \mathbb{R}^{10}$ , which influence aspects like height, weight, and limb ratios. Based on these parameters, the SMPL regression model estimates the triangular mesh vertices, denoted as  $v \in \mathbb{R}^{6890}$ , and the 3D joint positions, represented by  $s_{3D} \in \mathbb{R}^{24 \times 3}$ . Additionally, the global translation of a person can be captured using a parameter  $d \in \mathbb{R}^3$ . The parameter indicates the location information for positioning the SMPL model in 3D space, specifically denoting the 3D coordinates of the root joint.

Delving into the technical details, the feature vectors  $g^t$  and  $l^t$ , obtained from the Keypoint and Anchor Points Modules respectively, are concatenated. This combined feature vector is then transformed into SMPL parameters using FC layers, as represented by the equation  $[\theta^t, \beta^t, d^t] = \text{FC}([g^t, l^t]; w_p)$ , where  $w_p$  represents the weight parameters of the MLP. The mesh  $m^t = (v^t, s_{3D}^t)$  is subsequently derived from  $\theta^t, \beta^t, d^t$  using the pre-trained SMPL regression model [79], as  $m^t = \text{SMPL}(\theta^t, \beta^t) + d^t$ . To ensure the pre-trained model's integrity, which uniquely determines the mesh model from the pose and shape parameters, its weights are kept fixed during both the training and testing phases.

### 4.2.2 Loss Function

Our EventPointMesh network is trained using a comprehensive loss function designed to address key aspects such as pose, shape, translation, mesh vertex, and 3D joint positions.

**Pose Loss** ( $\mathcal{L}_{\text{pose}}$ ) measures the Mean Squared Error (MSE) between the predicted pose vector  $\theta_k^t$  and ground truth  $\hat{\theta}_k^t$  across all time-windows  $T$

and joints  $K$ .

$$\mathcal{L}_{\text{pose}} = \frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K \|\boldsymbol{\theta}_k^t - \hat{\boldsymbol{\theta}}_k^t\|_2^2, \quad (4.1)$$

**Shape Loss** ( $\mathcal{L}_{\text{shape}}$ ) measures the MSE between the estimated shape parameter  $\boldsymbol{\beta}^t$  and its corresponding ground truth  $\hat{\boldsymbol{\beta}}^t$  over all time-windows.

$$\mathcal{L}_{\text{shape}} = \frac{1}{T} \sum_{t=1}^T \|\boldsymbol{\beta}^t - \hat{\boldsymbol{\beta}}^t\|_2^2, \quad (4.2)$$

**Translation Loss** ( $\mathcal{L}_{\text{trans}}$ ) ensures accurate translation prediction by calculating the MSE between the predicted translation vector  $\boldsymbol{d}^t$  and ground truth  $\hat{\boldsymbol{d}}^t$ .

$$\mathcal{L}_{\text{trans}} = \frac{1}{T} \sum_{t=1}^T \|\boldsymbol{d}^t - \hat{\boldsymbol{d}}^t\|_2^2, \quad (4.3)$$

**Vertex Loss** ( $\mathcal{L}_{\text{vert}}$ ) aims to optimize the prediction of vertices by measuring the MSE of predicted vertices  $\boldsymbol{v}^t$  with the ground truth  $\hat{\boldsymbol{v}}^t$  across all time-windows and vertices  $P$ .

$$\mathcal{L}_{\text{vert}} = \frac{1}{TP} \sum_{t=1}^T \sum_{p=1}^P \|\boldsymbol{v}^t - \hat{\boldsymbol{v}}^t\|_2^2, \quad (4.4)$$

**3D Joint Loss** ( $\mathcal{L}_{\text{3D}}$ ) ensures accurate 3D joint position prediction by calculating the MSE of the predicted 3D joint positions  $\boldsymbol{s}_{\text{3D}}^t$  with their respective ground truth  $\hat{\boldsymbol{s}}_{\text{3D}}^t$ .

$$\mathcal{L}_{\text{3D}} = \frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K \|\boldsymbol{s}_{\text{3D}}^t - \hat{\boldsymbol{s}}_{\text{3D}}^t\|_2^2, \quad (4.5)$$

**2D Joint Loss** ( $\mathcal{L}_{\text{2D}}$ ) measures the discrepancy between the predicted heat-vector  $\boldsymbol{h}^t$  and its ground truth  $\hat{\boldsymbol{h}}^t$ . To compute this loss, the ground truth



3D joint positions  $\hat{s}_{3D}$  are first projected onto the 2D image plane. Using known camera parameters and the projection function  $\pi$ , the 2D joint coordinates  $\hat{s}_{2D} = (\hat{s}_x, \hat{s}_y)$  are obtained by  $\hat{s}_{2D} = \pi(\hat{s}_{3D})$ . Following the projection, a Gaussian filter is applied to generate the heat-vectors, which are represented as  $\hat{h}_x = [\hat{h}_0, \hat{h}_1, \dots, \hat{h}_i, \dots, \hat{h}_W]$  and  $\hat{h}_y = [\hat{h}_0, \hat{h}_1, \dots, \hat{h}_j, \dots, \hat{h}_H]$ . The confidence values for  $\hat{h}_x$  are calculated as:

$$\hat{h}_i = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(i - \hat{s}_x)^2}{2\sigma^2}\right). \quad (4.6)$$

The confidence values for  $\hat{h}_y$  can be computed similarly by replacing  $i$  with  $j$  and  $\hat{s}_x$  with  $\hat{s}_y$ . To quantify the divergence between the predicted and ground truth heat-vectors, the Kullback–Leibler (KL) divergence is employed:

$$\mathcal{L}_{2D} = \frac{1}{T} \sum_{t=1}^T \text{KL}(\mathbf{h}^t \parallel \hat{\mathbf{h}}^t). \quad (4.7)$$

**Overall Loss** ( $\mathcal{L}$ ) aggregates the aforementioned losses to train our Event-PointMesh network and is described by:

$$\begin{aligned} \mathcal{L} = & \lambda_{\text{pose}} \mathcal{L}_{\text{pose}} + \lambda_{\text{shape}} \mathcal{L}_{\text{shape}} + \lambda_{\text{trans}} \mathcal{L}_{\text{trans}} \\ & + \lambda_{\text{vert}} \mathcal{L}_{\text{vert}} + \lambda_{3D} \mathcal{L}_{3D} + \lambda_{2D} \mathcal{L}_{2D}. \end{aligned} \quad (4.8)$$

In this formulation, the  $\lambda$  coefficients serve as hyperparameters that balance the impact of each loss component during the model’s training phase. Notably, while mesh vertices  $\mathbf{v}$  and 3D joint positions  $\mathbf{s}_{3D}$  are uniquely determined by the pose parameters  $\theta$  and shape parameter  $\beta$ , the vertex loss  $\mathcal{L}_{\text{vert}}$  and the 3D joint loss  $\mathcal{L}_{3D}$  are included to enhance training stability and accelerate convergence.

### 4.2.3 EventPointMesh Dataset (EPMD)

**Why the Original Dataset is Required?** We propose our original dataset EPMD to address the lack of a dataset for HMR in low-light conditions,

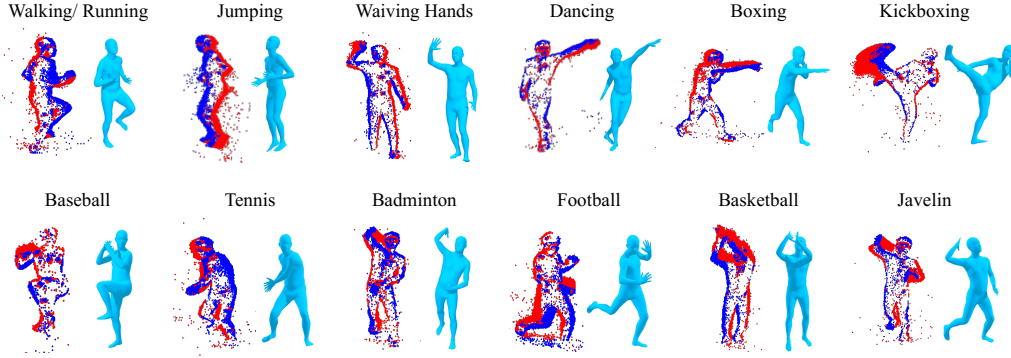


Figure 4.5: The EventPointMesh Dataset (EPMD) encompasses diverse actions from daily activities to sports motions. Subjects performed the actions multiple times in both well-lit and poorly-lit environments.

Table 4.1: Datasets for event-based human pose and shape estimation

| Dataset          | Seq/Sub | Frame | Pose | Shape | Poorly-lit Env. |
|------------------|---------|-------|------|-------|-----------------|
| DHP19 [10]       | 33/17   | 87k   | Yes  | No    | No              |
| Yelan-Syn [169]  | 8/10    | 4M    | Yes  | No    | Yes             |
| Yelan-Real [169] | 4/9     | 446k  | Yes  | No    | Yes             |
| EventCap [148]   | 2/6     | N/A   | N/A  | N/A   | N/A             |
| MMHPD [175]      | 12/15   | 240k  | Yes  | Yes   | No              |
| EPMD (Ours)      | 16/3    | 228k  | Yes  | Yes   | Yes             |

a domain where event cameras are particularly advantageous. Table 4.1 provides a comparison of EPMD with existing datasets for HPE and HMR, focusing on metrics such as the number of sequences per subject, total subjects, total frame count, as well as the availability of data on poses, shapes, and poorly-lit environments. EPMD presents mesh data acquired in varied lighting conditions, a feature not available in other HMR datasets such as EventCap [148] and MMHPD [175]. This enables the exploration of HMR in lighting conditions where conventional frame-based cameras struggle, highlighting the capabilities of event cameras. While the number of subjects in our dataset is fewer than existing datasets, EPMD offers a compa-

erable volume of captured data ranging from daily activities to sports motions, as illustrated in Fig. 4.5. This supports the creation of adaptable and broadly applicable HMR models, in contrast to datasets like Yelan-Syn and Yelan-Real [169], which are limited to dance movements. The dataset comprises not only mesh data but also synchronized intensity images, event data, and optical MoCap marker data. This multifaceted composition ensures EPMD is a valuable resource for advancing HMR research, particularly in challenging lighting conditions.

**Data Acquisition.** EPMD was acquired using two event cameras and an optical MoCap system. We used iniVation DAVIS346 models [9] with a spatial resolution of  $346 \times 260$  pixels and a temporal resolution of  $1 \mu\text{s}$  as event cameras. These cameras recorded synchronized grayscale frame images and event data, which were transmitted in packets. The event data within these packets were then converted into point cloud data segmented at fixed time intervals for input into the network. The event point clouds contain an average of 270k points per second across the entire dataset. The OptiTrack Motive [99] is used as the optical MoCap system. Three subjects were asked to wear a MoCap suit with markers and were recorded performing various motions. They performed medium-speed motions like walking, jogging, jumping, waving hands, and kicking, as well as fast motions such as fast running, dancing, boxing, kickboxing, baseball, tennis, badminton, football, basketball, and javelin (Fig. 4.5). These actions are captured in both bright and dark conditions. Our dataset includes approximately 228k intensity frames shot at about 15fps in bright conditions and about 4fps in dark conditions. It also contains event data synchronized with the intensity images, MoCap marker trajectory data captured at 120fps, and data from the generated SMPL mesh model, totaling about 3 hours. The study was formally approved by the Bioethics Committee of the Graduate School of Science and Technology at Keio University under approval number 2023-112.

**Annotation.** The annotation of the SMPL mesh model is conducted based on the trajectory data of the MoCap markers. By labeling the markers with their corresponding position names during recording with the OptiTrack, it becomes feasible to employ the Mosh++ [80, 90] method, which fits the SMPL model to the labeled marker data through optimization. The SMPL model generated by Mosh++ contains pose parameters, shape parameters, vertex coordinates of the mesh model, 3D joint positions, and 3D translation data in the MoCap coordinate system. A total of 24 joints are annotated, which include: Pelvis, Left and Right Hips, Left and Right Knees, Left and Right Ankles, Left and Right Feet, Spine1, Spine2, Spine3, Neck, Head, Left and Right Collars, Left and Right Shoulders, Left and Right Elbows, Left and Right Wrists, and Left and Right Hands. The ground truth data of 2D joint positions used in the Keypoints Module is acquired by projecting the 3D joint positions onto the image in the event camera coordinate system. The intrinsic camera parameters used in this projection are obtained through calibration using a checkerboard pattern. The extrinsic parameters are calculated by obtaining corresponding points in the MoCap and event camera coordinate systems and solving the Perspective-n-Point (PnP) problem.

## 4.3 Experiments

### 4.3.1 Experimental Settings

**Dataset.** To demonstrate the efficacy of our approach, we employed both the existing large-scale dataset MMHPSD [175] and our EPMD for experiments. The MMHPSD comprises data from a total of 15 subjects: 11 males and 4 females. Each subject performed a total of 21 different motions, categorized into three groups based on speed: fast, middle, and slow. Each motion was repeated four times. For each subject, 12 video clips (totaling 180

clips) were collected, each consisting of approximately 1,300 frames spanning about a minute and a half at 15fps. Additionally, our EPMD uniquely contains a rich set of data recorded in low-light conditions, which are not found in existing datasets for HMR. Therefore, we used the EPMD to assess the low-light robustness of our proposed method.

**Implementation Details.** During training, the event data was segmented by time windows corresponding to the intervals of the intensity image frames and treated as point cloud data between adjacent frames. Given the vast volume of raw data points, which posed challenges in storage and data processing, the point cloud within each time window was randomly sampled down to 7,500 points to reduce data size. The sequence of event data fed into the network was time series point cloud data for 15 consecutive time windows. In the Keypoints Module, we identified 13 primary human body joints (Pelvis, Left and Right Knee, Left and Right Ankle, Spine, Head, Left and Right Shoulder, Left and Right Elbow, and Left and Right Wrist) and estimated their 2D joint positions following Xue *et al.* [151]. In the Anchor Points Module, we grouped the nearest 500 points from the spatiotemporal domain based on the 2D Euclidean distance on the image plane, centered around the 2D joint coordinates. For network training, we employed the Adam optimizer [63]. The weights for each element in the loss function, namely  $\lambda_{\text{pose}}$ ,  $\lambda_{\text{shape}}$ ,  $\lambda_{\text{trans}}$ ,  $\lambda_{\text{vert}}$ ,  $\lambda_{3D}$ , and  $\lambda_{2D}$ , were empirically determined to be 10, 10, 10, 100, 10, and 1, respectively. To discourage settling into local minima during the learning process, the learning rates for the Base Module and Keypoints Module were set to  $1e-4$ , while that of the Anchor Points Module and SMPL Module were set to  $1e-3$ , based on experimental findings. The batch size was set to 8. End-to-end learning was conducted across the entire network, including all modules.

**Evaluation Metrics.** Referring to the prior research [175], we evaluated experimental results using six metrics: mean per joint position error (MPJPE),

pelvis-aligned MPJPE (PEL-MPJPE), Procrustes-aligned MPJPE (PA-MPJPE), per vertex error (PVE), percentage of correct keypoints (PCKh@0.5), and miss rate. In Table 4.2 and Table 4.4, MPJPE, PA-MPJPE, PEL-MPJPE, and PVE are all denoted in mm.

MPJPE calculates the average 3D Euclidean distance between the ground truth and estimated joint positions. PEL-MPJPE computes MPJPE after aligning the translation of the root joint. PA-MPJPE is an evaluation metric that calculates MPJPE after applying a Procrustes analysis [32], which aligns the predicted and ground true poses by minimizing the distances between corresponding joints through translation, scaling, and rotation adjustments. PVE metric assesses the 3D Euclidean distance between the true and estimated coordinates of each vertex of the SMPL mesh model. MPJPE and PVE are respectively computed using the following equations.

$$\text{MPJPE} = \frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K \|s_{3D_t}^k - \hat{s}_{3D_t}^k\|_2, \quad (4.9)$$

$$\text{PVE} = \frac{1}{TP} \sum_{t=1}^T \sum_{p=1}^P \|q_t^p - \hat{q}_t^p\|_2, \quad (4.10)$$

In the above equation,  $T$ ,  $K$ ,  $P$ ,  $s_{3D}$ , and  $q$  represent the number of time windows for the event point clouds, the number of body joints, the number of vertices in the SMPL mesh model, the 3D coordinates of the joints, and the vertex coordinates of the SMPL mesh model, respectively.  $\hat{s}_{3D}$  and  $\hat{q}$  represent the ground truth corresponding to the estimated values  $s_{3D}$  and  $q$ . PCKh@0.5 evaluates the percentage of joints whose Euclidean distance between the ground truth and the estimated position is less than 50% of the bone length from the neck to the head. Miss rate evaluates the percentage of failure to output a mesh in the test data.

Table 4.2: Quantitative results from experiments using the MMHPSD and EPMD datasets.

| Method                                | MPJPE<br>[mm] ( $\downarrow$ ) | PEL-MPJPE<br>[mm] ( $\downarrow$ ) | PA-MPJPE<br>[mm] ( $\downarrow$ ) | PVE<br>[mm] ( $\downarrow$ ) | PCKh@0.5<br>[%] ( $\uparrow$ ) | Miss Rate<br>[%] ( $\downarrow$ ) |
|---------------------------------------|--------------------------------|------------------------------------|-----------------------------------|------------------------------|--------------------------------|-----------------------------------|
| <b>MMHPSD Dataset (All-Subject)</b>   |                                |                                    |                                   |                              |                                |                                   |
| EventHPE(GT)                          | <u>72.3</u>                    | <u>52.6</u>                        | <u>41.3</u>                       | <u>50.5</u>                  | <u>86.0</u>                    | -                                 |
| EventHPE(VIBE)                        | -                              | 71.3                               | 51.4                              | (70.5)                       | 79.6                           | 1.4                               |
| EPM(Ours)                             | 97.4                           | <b>59.1</b>                        | <b>42.8</b>                       | <b>51.9</b>                  | <b>84.8</b>                    | <b>0.0</b>                        |
| <b>MMHPSD Dataset (Cross-Subject)</b> |                                |                                    |                                   |                              |                                |                                   |
| EventHPE(GT)                          | <u>103.9</u>                   | <u>80.4</u>                        | 64.8                              | 79.0                         | <u>76.0</u>                    | -                                 |
| EventHPE(VIBE)                        | -                              | 88.5                               | 66.6                              | (86.0)                       | 69.3                           | 1.4                               |
| EPM(Ours)                             | 145.1                          | <b>86.3</b>                        | <b>61.6</b>                       | <b>74.7</b>                  | <b>71.9</b>                    | <b>0.0</b>                        |
| <b>EPMD Dataset (All-Subject)</b>     |                                |                                    |                                   |                              |                                |                                   |
| EventHPE(GT)                          | 198.0                          | 90.3                               | 71.2                              | 81.1                         | 70.2                           | -                                 |
| EventHPE(VIBE)                        | -                              | 128.9                              | 95.7                              | (109.0)                      | 55.0                           | 2.1                               |
| EPM(Ours)                             | <u>110.7</u>                   | <b>86.0</b>                        | <b>60.7</b>                       | <b>71.8</b>                  | <b>71.7</b>                    | <b>0.0</b>                        |

### 4.3.2 Comparison Against Event-based Methods.

Through three experiments utilizing two distinct datasets, MMHPSD and EPMD, we demonstrate the efficacy of our approach in estimating meshes solely from event data compared to baseline methods. The experiments were conducted under two different settings: (i) an all-subject setting, where data for all subjects was divided into training and test sets, and (ii) a cross-subject setting, where different subjects' data were used for training and testing. Given the large number of subjects included in the MMHPSD, the experiment was conducted under both conditions (i) and (ii). On the other hand, EPMD, designed for assessing performance under diverse lighting conditions, contains fewer subjects; hence, only condition (i) was applied in its experiments. This subsection first details the baseline methods, followed by a description of the experiments conducted using each dataset.

**Baseline Methods.** Our approach is the first to estimate the 3D human meshes solely from the point cloud of event data, and no existing methods

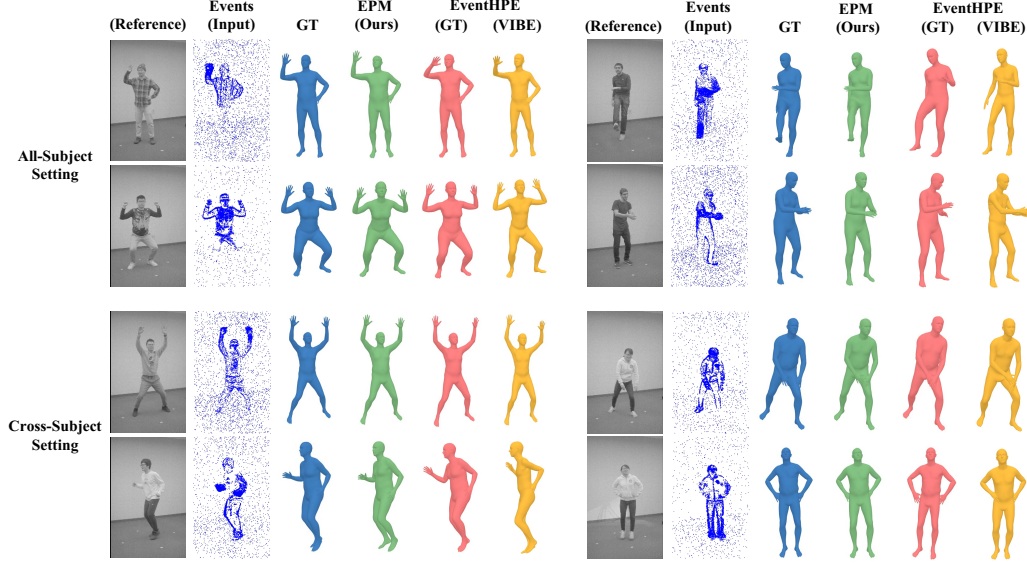


Figure 4.6: Qualitative results from the experiment using MMHPSD as detailed in Sec 4.3.2. The top rows display the results under the all-subject setting, while the bottom rows show the results under the cross-subject setting. Although our proposed method only uses event data as input, it estimates the mesh with an accuracy comparable to EventHPE(VIBE), which uses intensity images to estimate the initial mesh for each sequence, and EventHPE(GT), which employs ground truth mesh.

are tackling the same task. Therefore, we compared our network model against EventHPE [175], a similar approach to ours. EventHPE differs significantly from our method in that it requires intensity images in addition to event frames to obtain the initial mesh model at the starting point for the sequence. Subsequent mesh models in the sequence are then estimated based on their displacements from this initial model. EventHPE computes this initial mesh model by using VIBE [65], which estimates the SMPL mesh from videos.

Throughout the experiments, EventHPE using VIBE is denoted as EventHPE(VIBE), and the method using the ground truth mesh as the initial mesh of the sequence is referred to as EventHPE(GT). Following the ex-



perimental setup of Zou et al. [175], EventHPE(GT) operates under the assumption that the mesh of the initial frame in each sequence is known, and the sequence consists of 8 frames. This method represents the upper bound of estimation accuracy that EventHPE network can achieve. In Table 4.2, the highest accuracy values, when including EventHPE(GT), are underlined, while the best scores excluding it are highlighted in bold. The VIBE [65] estimates camera parameters based on a weak perspective projection model, not accounting for global translation, and thus we do not report MPJPE for EventHPE(VIBE). Additionally, the mesh model used by VIBE is a gender-neutral model, unlike the gender-specific models used in MMHPSD and EPMD. Consequently, the PVE values for EventHPE(VIBE) are provided in parentheses as reference values.

**Evaluation on MMHPSD (All-Subject).** We evaluated the generalization ability of the models under the all-subject setting. Specifically, the networks were trained using three out of the four video sequences representing each subject’s actions and used the remaining sequence for testing. The qualitative and quantitative results are presented in Table 4.2 and the upper part of Fig. 4.6, respectively. A comparison of the methods, as shown in Fig.4.6, reveals that despite slight discrepancies in body orientation and limb positions, the models generally achieve mesh estimations of high accuracy that closely mirror the ground truth. A detailed examination of Table4.2, which presents results across the entire test dataset, indicates that EventHPE(GT) with ground truth-based initialization achieves the highest accuracy. Without EventHPE(GT), our method EPM(Ours) emerged as the most accurate. This underscores that our approach, despite not relying on intensity images, outperforms existing methods that estimate meshes from intensity images and event data.

**Evaluation on MMHPSD (Cross-Subject).** We aimed to assess the generalization performance of the models across different subjects. We con-

ducted a cross-subject evaluation by training our network on data from 12 out of the 15 subjects and employing the remaining 3 subjects' data for testing. The qualitative and quantitative results are shown in Table 4.2 and the lower part of Fig. 4.6, respectively. Our method consistently surpassed EventHPE(VIBE) in all metrics, and the results for PEL-MPJPE and PVE were even better than those for EventHPE(GT). PEL-MPJPE and PVE evaluate the error in 3D joint coordinates when aligning translation and rotation of the root joint. These outcomes suggest that our method excels in estimating the pose and shape relative to the root joint from event data compared to that of EventHPE [175]. However, the influence of root joint rotation on accuracy was evident, indicating room for improvement.

Comparing the outcomes for the all-subject setting and the cross-subject setting revealed that all metrics degraded in the cross-subject setting, hinting at a reduced estimation accuracy when predicting non-included individuals. This suggests that variations in the same movements by different subjects, along with differences in clothing and physique, led to distinct event data characteristics. Future work could focus on developing techniques to mitigate the effects of clothing and physique variations on data characteristics.

**Evaluation on EPMD (All-Subject).** To evaluate the robustness of the methods against variations in lighting, we conducted an all-subject evaluation using EPMD that contains data captured while subjects performed various medium to high-speed movements in both well-lit and poorly-lit environments. Due to the limited number of subjects, in this experiment, we utilized 80% of the takes from each subject's data for training and the remaining 20% for testing. The experimental results are presented in Fig. 4.7 and Table 4.2. The optical flows, displayed in Fig. 4.7, were estimated by the FlowNet module used in EventHPE. It is important to note that in our experiments, this optical flow is commonly utilized in both EventHPE(GT)

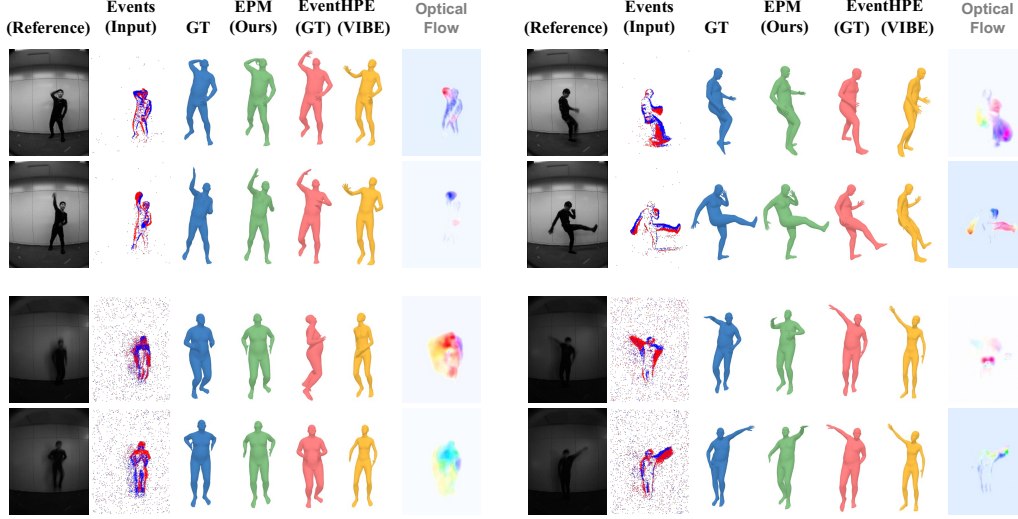


Figure 4.7: Qualitative results from the experiment using EPMD under the all-subject setting (Sec 4.3.2). The top two rows display well-lit data results, while the bottom two rows show poorly-lit data results. The images, segmented into four blocks separated by wide spaces, are adjacent frames arranged in the top and bottom rows. Note that the Optical Flow, assisting mesh estimation in EventHPE(VIBE) and EventHPE(GT), comes from the FlowNet module and is unrelated to EMP (Ours).

and EventHPE(VIBE). In contrast, our proposed method, EPM(Ours), does not make use of optical flow.

As can be seen from Table 4.2, our method achieved the highest accuracy across all metrics, even when including EventHPE(GT). While the EventHPE [175] inputs not only the event frames but also the optical flow estimated from them into the mesh estimation network ShapeNet, its performance tends to deteriorate when the optical flow estimation is not accurate, as illustrated in the lower part of Fig. 4.7. In particular, in the bottom right two rows showcasing the EventHPE method, both adjacent frames appear to be in almost the same pose. This is due to EventHPE’s mechanism of using a sequence of data comprising optical flow and event frames from eight adjacent frames for mesh estimation. If the sequence includes

an optical flow that has significantly failed in estimation, it greatly impacts the accuracy of subsequent frame estimations, resulting in a failure to replicate changes in the subject's movement.

One potential cause for these inaccuracies in optical flow estimation lies in the presence of motion blur in the intensity images and the reduced frame rates. When capturing data in dim lighting, the camera requires longer exposure times to gather sufficient light, leading not only to motion blur but also to reduced frame rates. This reduction in frame rate means that the poses between adjacent frames can vary significantly. Since EventHPE trains its FlowNet to predict optical flow based on such intensity images, both the motion blur and the large pose differences between frames can hinder accurate optical flow estimation. This adverse effect impacts not just the EventHPE [175] network but also VIBE [65], which is utilized to predict the mesh of the sequence's initial frame. In contrast, EventPointMesh can estimate meshes with high accuracy without being affected by issues related to intensity images.

The miss rate in Table 4.2 indicates the proportion of test data frames where the mesh was not estimated. EventHPE(VIBE) failed to estimate the mesh for 1.4% of the frames in MMHPSD and 2.1% in EPMD, resulting in missing outputs. In contrast, our method, which solely relies on event data without using intensity images, achieves the miss rate of 0.0% as it is not subject to the adverse effects of the intensity image caused by the dark environment. The miss rate of 0.0% represents an absolute zero value, indicating that our proposed method reliably outputs a mesh model based on the input event data. The presence of missing parts in the estimated mesh could necessitate additional processing, such as motion infilling, to fill in the gaps, particularly in applications requiring online streaming. This could potentially affect real-time performance. From this perspective, the error rate of 0.0% for our proposed method is an important and beneficial metric for the development of future applications.

Table 4.3: Quantitative Results of the Comparison between Event-based and Frame-based Methods.

| Method    | PEL-MPJPE<br>[mm] ( $\downarrow$ ) | PA-MPJPE<br>[mm] ( $\downarrow$ ) | PVE<br>[mm] ( $\downarrow$ ) |
|-----------|------------------------------------|-----------------------------------|------------------------------|
| VIBE(L)   | 150.4                              | 75.6                              | 89.7                         |
| VIBE(D)   | 212.9                              | 123.8                             | 145.7                        |
| VIBE(L+D) | 158.9                              | 82.1                              | 97.3                         |
| EPM(L)    | 84.4                               | 59.4                              | 70.0                         |
| EPM(D)    | 103.3                              | 75.0                              | 90.2                         |
| EPM(L+D)  | 86.0                               | 60.7                              | 71.8                         |

### 4.3.3 Comparison Against a Frame-based Method

We demonstrate the effectiveness of our method utilizing an event camera under conditions that are challenging for traditional frame-based cameras, namely dim environments where images tend to become dark, unclear, and prone to motion blur. To this end, we conduct accuracy comparison experiments between the existing frame-based camera method, VIBE [65], and our proposed method, EventPointMesh. We divided the EPMD dataset into data captured in light and dark environments and tested both VIBE and EventPointMesh against each setting. The event camera used in this study dynamically adjusts exposure time according to lighting conditions to obtain images with sufficient brightness. During our data collection, the camera captured at approximately 15fps in light environments and around 4fps in dark environments. Consequently, the number of frames was 188k in the light environment and 40k in the dark environment, leading to an imbalance in the number of frames between the two conditions. However, since the subjects performed the same actions for the same duration under both lighting conditions, the data is balanced in terms of the types and frequency of actions performed by the subjects across both conditions. Tests conducted in light environments are denoted

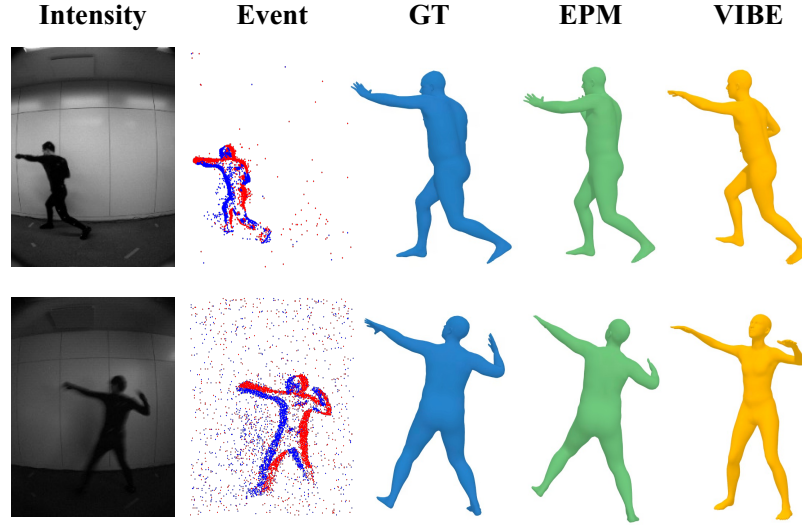


Figure 4.8: Qualitative results of the experiment comparing our event-based method with the frame-based method. The top row displays the mesh estimation results of both methods in bright environments, while the bottom row presents the estimation results in dark environments.

as VIBE(L) and EPM(L), while those in dark environments are marked as VIBE(D) and EPM(D). Tests on data from both lighting environments combined are labeled as VIBE(L+D) and EPM(L+D). The results of these experiments are presented in Table 4.3. Additionally, qualitative results are shown in Fig. 4.8.

The results from Table 4.3 demonstrate that our proposed method achieves higher accuracy across all conditions, whether in light, dark, or mixed environments. Such a consistent performance gap is considered to stem from differences in the ability to represent temporal information between frame-based and event-based methods. While frame-based approaches rely solely on static appearance information observed at discrete frame timestamps, EventPointMesh exploits events occurring within the same time interval, enabling it to capture whether motion occurs within short durations as well as the magnitude of such changes. Furthermore, com-

paring the relative increase in error from light (L) to dark (D) environments between VIBE and EPM reveals that VIBE experiences a larger increase in error. For instance, while EPM shows an increase of +22.4% in PEL-MPJPE and +26.3% in PA-MPJPE, VIBE’s errors escalate to +41.6% in PEL-MPJPE and +63.8% in PA-MPJPE. This indicates that the frame-based method VIBE is more adversely affected by dim lighting. Moreover, comparing the increase in the magnitude of PEL-MPJPE relative to PA-MPJPE in dark environments (D), EPM increases by +37.7%, whereas VIBE significantly jumps to +72.0%.

This disparity suggests that VIBE, when estimating meshes in dark environments, often misestimates the orientation of the person, confusing the front and back. The top row of Fig. 4.8 presents the estimation results of EPM and VIBE in light environments, while the bottom row shows the results in dark environments. It is evident in the dark environments that VIBE reverses the orientation of the person compared to the ground truth mesh. This phenomenon leads to a relatively larger error in PEL-MPJPE, which calculates error after aligning the root joint’s position only, compared to PA-MPJPE, which aligns position, rotation, and scale for error calculation. The results from both quantitative and qualitative analyses indicate that VIBE, which utilizes traditional frame-based cameras, experiences a significant degradation in estimation accuracy in dark environments. In contrast, EventPointMesh, which solely relies on event data, shows a minor decrease in estimation accuracy. These findings demonstrate the high robustness of our proposed method against dim lighting conditions.

#### 4.3.4 Ablation Study

We conduct an ablation study to verify the contribution of our coarse-to-fine feature extraction to the accuracy of mesh estimation. The architecture of our EPM network leverages the global features from the Keypoints

Table 4.4: Quantitative results from the ablation study

| Method   | MPJPE<br>[mm] ( $\downarrow$ ) | PEL-MPJPE<br>[mm] ( $\downarrow$ ) | PA-MPJPE<br>[mm] ( $\downarrow$ ) | PVE<br>[mm] ( $\downarrow$ ) | PCKh@0.5<br>[%] ( $\uparrow$ ) |
|----------|--------------------------------|------------------------------------|-----------------------------------|------------------------------|--------------------------------|
| EPM(G)   | 115.8                          | 89.3                               | 63.2                              | 73.3                         | 69.9                           |
| EPM(L)   | 116.0                          | 87.0                               | 61.8                              | 71.9                         | 71.0                           |
| EPM(G+L) | <b>110.7</b>                   | <b>86.0</b>                        | <b>60.7</b>                       | <b>71.8</b>                  | <b>71.7</b>                    |

Module and the local features from the Anchor Points Module. Both these features are integrated within the SMPL Module. To illustrate the contributions of these elements to mesh estimation, we evaluate the accuracy of estimation when each component is omitted. We denote the method that estimates meshes using only global features as EPM(G), the method using only local features as EPM(L), and the method leveraging both as EPM(G+L). The results of the experiments are presented in Table 4.4.

The results indicated that EPM(G+L) achieved the highest scores across all metrics. When comparing EPM(G) and EPM(L), it was found that EPM(G) had higher accuracy in MPJPE, while EPM(L) scored higher in both PEL-MPJPE and PA-MPJPE. MPJPE is a metric that evaluates the 3D Euclidean distance of joint coordinates without aligning the estimated and ground true poses in terms of position, rotation, and scale. Therefore, it is reasonable that EPM(G), which captures the coarse global features of human pose, 3D position, and body shape, performed better in MPJPE. On the other hand, PEL-MPJPE applies MPJPE after aligning the position of the root joint, and PA-MPJPE applies MPJPE after aligning translation, rotation, and scale. The higher accuracy of EPM(L) in these metrics suggests that focusing on the finer details around each joint led to better precision than EPM(G). As global and local features can be complementary, the combined use of both in EPM(G+L) resulted in further improved accuracy over either approach alone. Therefore, our pioneering coarse-to-fine feature extraction approach, initially extracting coarse global features of the event point





Figure 4.9: Qualitative results of the Task2 described in Sec. 4.3.5. The topmost “EPM(img)” represents our method’s results when segmenting events according to the intensity image frame rate, which varies with the lighting conditions. The middle “EPM(120)” shows the outcomes when segmenting events and estimating the meshes at the same 120fps. The bottom “GT” shows the ground truth meshes with a frame rate of 120fps. Note that the transparent meshes in the bottom two rows are only for visual clarity; there is no difference from the opaque meshes.

cloud and then grouping the spatiotemporal event point cloud around each joint to extract detailed local features, demonstrates effectiveness in HMR using only event data.

### 4.3.5 Evaluation Under Various Input Conditions

**Task1: Different length of input event sequence.** This experiment was conducted to evaluate the effectiveness of feeding the event point cloud into the network as sequential temporal data. In our EPM network, we incorporate BiLSTM into both the Keypoints Module and Anchor Points Modules to consider the temporal characteristics of the input event point clouds. We demonstrate the efficacy of the proposed network architecture that accounts for temporal features by evaluating the accuracy of the estimated mesh from varying lengths of event data. In this study, the event

Table 4.5: Quantitative evaluation under various input conditions.

|       | Method   | MPJPE<br>[mm] ( $\downarrow$ ) | PEL-MPJPE<br>[mm] ( $\downarrow$ ) | PA-MPJPE<br>[mm] ( $\downarrow$ ) | PVE<br>[mm] ( $\downarrow$ ) | PCKh@0.5<br>[%] ( $\uparrow$ ) |
|-------|----------|--------------------------------|------------------------------------|-----------------------------------|------------------------------|--------------------------------|
| Task1 | EPM(1)   | 167.3                          | 138.1                              | 97.2                              | 111.0                        | 51.7                           |
|       | EPM(5)   | 125.2                          | 100.8                              | 70.5                              | 82.4                         | 65.8                           |
|       | EPM(15)  | <b>110.7</b>                   | <b>86.0</b>                        | <b>60.7</b>                       | <b>71.8</b>                  | <b>71.7</b>                    |
| Task2 | EPM(120) | 160.6                          | 129.3                              | 90.6                              | 105.6                        | 54.8                           |
|       | EPM(img) | <b>110.7</b>                   | <b>86.0</b>                        | <b>60.7</b>                       | <b>71.8</b>                  | <b>71.7</b>                    |

point clouds fed into the network were divided into individual time windows between intensity image frames. We experimented with three patterns: feeding event point clouds from 1, 5, and 15 continuous time windows into the model, denoted as EPM(1), EPM(5), and EPM(15) respectively. The experimental results are shown in Table 4.5.

The results indicate that EPM(15) yielded the highest mesh estimation accuracy, demonstrating the effectiveness of extracting temporal features for mesh estimation. Due to the event cameras' nature, they don't capture static parts of the subject. This makes estimating the entire body mesh from brief event data segments, such as a single time window, challenging. However, our proposed network overcomes these limitations, enabling high-precision mesh estimation.

**Task2: Different width of time-window.** We evaluate the precision of mesh estimation over different time-window widths, assessing how adaptably the EventPointMesh network can handle variations in time-window sizes. In previously discussed experiments, event data were segmented according to the frame rate of intensity images captured by the event camera. The frame rate of intensity images within the EPMD is roughly 15fps under well-lit conditions and about 4fps in poorly-lit scenarios. Prior research, EventHPE [175], estimated the mesh synchronized to the frame rate of the

intensity images taken with the event camera. This approach inherently limits the maximum frame rate of the output mesh to that of the intensity images. In contrast, our EventPointMesh does not rely on intensity images, allowing it to process input event data at any chosen time window. Consequently, we carried out training and testing with event data segmented to match the 120fps time window, corresponding to the optical MoCap [99] data.

The experimental results are presented in Figure 4.9 and Table 4.5. The method segmented based on the frame rate of intensity images is labeled as EPM(img), while the one conducted at 120fps is defined as EPM(120). The results show that although EPM(120) is slightly less effective than EPM(img), it still possesses sufficient capability for mesh estimation. The decrease in accuracy for EPM(120) compared to EPM(img) can be attributed to the significantly reduced amount of event information in the time window when operating at a high frame rate and the subject's movements are slow. In this experiment, the number of events per time window is set to 7500, and if the detected number of events falls below this, points are randomly duplicated to increase to 7500, but this does not add useful information for reconstructing the movement. Therefore, at this stage, increasing the frame rate reduces the number of events per time window, thereby decreasing the estimation accuracy. Addressing this issue by developing a mechanism that can extract useful information from a small number of events, rather than randomly duplicating them, could lead to the construction of a higher frame rate method.

Figure 4.9 demonstrates that our proposed method can achieve HMR at a high frame rate comparable to traditional optical motion capture (MoCap). This indicates that even in challenging environments for traditional frame-based cameras, such as poorly-lit conditions like dark rooms and high-speed subject movements, our method enables a simplified 3D HMR with just a single monocular event camera.

## 4.4 Discussion and Limitations

### 4.4.1 Failure Cases

Our method has liberated event-based HMR from dependence on intensity images, enhancing its applicability in real-world scenarios. Moreover, although our approach involves bidirectionality in its processing, it can sequentially estimate meshes from event point clouds, making it feasible for transition to online applications. Specifically, there is potential for utilizing our method in activity monitoring systems within environments requiring privacy, such as hospitals, care facilities, and homes. Additionally, for existing efforts in developing projection mapping that projects video fitted to the body of performers on stage in real-time, our method offers a highly beneficial system capable of handling fast movements in dark environments. However, towards the realization of such applications, there are several scenarios where we currently face challenges, necessitating improvements. Below, we discuss several representative failure cases and the corresponding strategies for addressing them.

**Case 1: Few or no detected events.** This case considers three types of scenarios. The first and most common scenario is when there is minimal movement of the body. Event cameras function by detecting per-pixel changes in luminance. Thus, movements that are not significant but still result in changes in luminance values will be detected as events. Since EventPointMesh utilizes time-series point cloud data as input, it is capable of estimating the most plausible pose for the entire body by leveraging the features from event point clouds of other moving body parts, even if events for a specific part of the body are temporarily sparse. This capability is evidenced in Fig. 4.6, where accurate mesh estimations are achieved even in instances where a body part is not captured in the events.

However, with the current network architecture, if the 2D joint positions estimated by the Keypoint Module are incorrect, it can result in the inability to extract useful information from the point cloud for mesh estimation in that local area, or it might capture features that adversely affect the estimations of subsequent modules. One effective solution to this problem could involve modifying the loss function used in the Keypoint Module for estimating 2D joint positions from KL divergence to Negative Log-Likelihood. This adjustment would allow the estimation process to also yield confidence levels, which could then be used by the Anchor Points Module to more accurately extract local features. Consequently, this could enhance the transmission of only the most beneficial local features to subsequent modules. The potential benefits of this refinement will be explored in future research to develop the method further.

Nevertheless, the current system, which utilizes BiLSTM to account for temporal relationships, can result in the output of meshes with high uncertainty in situations where there is a prolonged static state, as it may lead to the loss of information supporting the presence of certain body parts. An effective solution to such long-term event omissions could be the implementation of autoregressive network architectures. An autoregressive network incorporates a loop structure that feeds the pose information, the output of the current time window, combined with the event data for the next time window as input to estimate the pose for the next time window. This means that once the pose has been estimated from the events associated with movement of body parts, the existence of these parts can still be inferred in their last known positions even if no new events are generated due to their lack of movement. Hence, it is anticipated that accurate estimations of the entire body pose at any given moment can be made based on the pose information carried over from previous time windows, even in cases where no events are detected over an extended period.

When designing such an autoregressive network, an important consideration is computational efficiency, particularly in long-horizon inference settings. In this study, event data are processed using fixed temporal windows. However, given the characteristics of event cameras, a uniform time-based processing scheme is not necessarily optimal. For example, alternative strategies could include processing data when a certain number of events have been accumulated, or dynamically adjusting the amount of computation within fixed time intervals by simplifying processing when only a small number of events are present. Such adaptive processing mechanisms, which respond to the density of incoming event data, have the potential to reduce redundant computations and improve overall computational efficiency and inference speed.

The second scenario involves situations where parts or the entirety of the subject's body move out of the camera's field of view, a common occurrence in real-world applications utilizing video-based pose estimation methods. The EPMD dataset assumes that the subject remains within the camera's field of view, with almost all data capturing the full body. Consequently, the pose estimation accuracy decreases in this scenario due to the network being trained on such data. When only a small part of the body (such as the hands, feet, or head) moves out of view, the system can still estimate a plausible 3D body model that includes the body parts located outside the 2D image frame by utilizing temporal point cloud features and the structural constraints of the parametric human model. However, as the amount of the body outside the field of view increases, the accuracy of position and pose estimation deteriorates, ultimately leading to an unexpected output as illustrated in the upper part of Fig. 4.10. Ideally, no mesh model should be generated when the subject is outside the camera's view, as no events occur within the frame. However, the current system produces a standing pose mesh, likely because the model outputs the most common pose within the dataset when faced with such an unfamiliar sce-

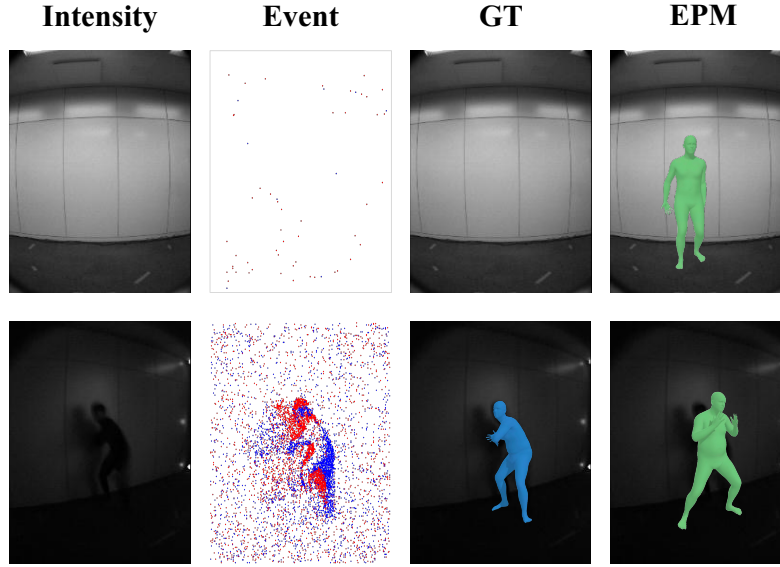


Figure 4.10: Failure cases of mesh estimation by EventPointMesh. The top row presents the output of our method for a scenario discussed in 4.4.1 Case 1, where the subject steps out of the camera’s view. Currently, in cases where no events are generated due to the absence of the subject, a mesh in a typical standing pose is outputted. The bottom row shows an example of mesh estimation when events are generated by parts other than the subject, as discussed in 4.4.1 Case 2. It demonstrates that the presence of shadows causing events can lead to a reduction in mesh estimation accuracy.

nario. To address this issue, incorporating a mechanism commonly known as the “top-down approach” in RGB image-based pose estimation methods, which estimates poses based on bounding boxes detected as human regions, could effectively handle situations where the subject moves in and out of the field of view.

The third scenario occurs when insufficient lighting prevent the generation of events. The EPMD dataset includes event data captured under both well-lit and poorly-lit lighting conditions. In Fig. 4.7, event data captured in dim environments appears to have more background noise compared to that from well-lit environments at first glance. However, this appear-

ance is due to accumulating event point clouds over long time windows to match the frame rate of intensity images, creating a single event frame. Thus, event cameras fundamentally are not adversely affected by dark environments and can accurately capture the movements of subjects with high temporal resolution without blurring. Nonetheless, poor lighting can impact the accuracy of event data. This issue arises in lighting conditions so dim that the changes in luminance caused by the subject's movement do not exceed the threshold set on the event camera device for detecting event occurrences. In such extremely dark environments, it might be preferable to use night vision cameras utilizing infrared instead of event cameras.

**Case 2: Events caused by entities other than the subject.** Our method faces challenges even when events are generated by entities other than a single subject, such as multiple people within the frame or movements of the camera itself. EPMD dataset is specifically designed for a stationary event camera capturing a single subject. When other humans or dynamic objects enter the field of view, or the camera moves, events unrelated to the subject's movement are detected, which can significantly compromise the accuracy of mesh estimation. Indeed, some data in the EPMD dataset captured in dim environments show shadows cast on the wall behind the subject. The bottom row of Fig. 4.10 presents an example where such conditions have led to a decrease in estimation accuracy. As illustrated by the intensity and event images in the figure, the presence of a shadow on the left side of the subject can disrupt event patterns to an extent that currently impacts mesh estimation accuracy. Therefore, the addition of other people's movements or camera motion is expected to further reduce estimation accuracy. One potential solution to this problem is to enrich the dataset with real or synthesized data capturing scenes with multiple humans or objects, or where the camera is in motion. Alternatively, introducing detectors and trackers to segment event data by individual or object and estimating the intended subject's pose could be an effective approach.



#### 4.4.2 Room for Improvement in Event-based HMR

Existing event-based HMR methods, including this study, still leave an important aspect unaddressed: the ability to accurately represent the clothing worn by the subjects. As demonstrated by the results of the experiment in the Sec. 4.3.2 “Evaluation on MMHPSD (Cross-Subject)”, differences in clothing and individual subject movements lead to variations in events, which in turn affect the accuracy of mesh reconstruction. This issue becomes more pronounced in EPMD, where the limited number of subjects worsens the problem. Addressing this challenge necessitates the creation of more diverse datasets through the augmentation of subject numbers and the use of data synthesis with computer graphics in future work.

In addition to this issue, there are more fundamental problems that require deeper solutions. Both the MMHPSD and EPMD datasets utilize the SMPL model to replicate the body shape including clothing. However, models trained on such datasets face limitations in generalizing across diverse clothing types when applied in real-world settings. While MMHPSD attempts to fit the SMPL model to the 3D shapes reconstructed using multiple RGB-D cameras around subjects in various outfits, this approach struggles with accurately representing oversized clothes or skirts. Similarly, EPMD employs optical MoCap to capture motion data, necessitating the use of specialized suits for attaching optical markers. Consequently, the placement and number of markers can alter the shape of the SMPL model, also presenting challenges in accommodating diverse clothing styles.

While this issue is also present in HMR using traditional frame-based cameras, it is particularly pressing for event cameras due to their principle of capturing only the subject’s movement. The fluctuation of clothing has a significant impact on the captured event data, demanding more thorough solutions. In future research, one potential solution could involve constructing mesh models that differentiate between the subject’s body

shape and the shape of their clothing. Additionally, developing mechanisms capable of distinguishing whether captured events are caused by body movement or clothing movement will also be crucial for achieving more accurate HMR.

## 4.5 Summary

This chapter explored the potential of using event data exclusively for effective human mesh recovery (HMR), bypassing the limitations of conventional intensity images. Our proposed EventPointMesh framework demonstrated superior performance by interpreting event data as a 3D spatio-temporal point cloud and leveraging both global and local features. The series of experiments and the ablation study affirmed the efficacy of our approach, especially under varied lighting conditions. However, several challenges remain, such as capturing static body parts in event data, people entering and exiting the frame, and events caused by objects other than the subject. Future endeavors will focus on enhancing accuracy and adaptability in real-world scenarios, but our current achievements represent a significant leap in event-based HMR, paving the way for efficient methodologies in challenging conditions.

While event-driven sensing offers clear robustness advantages, selecting the appropriate sensing modality ultimately depends on the target application. In many everyday scenarios, long-term monitoring or privacy considerations may render vision sensors less suitable. In such cases, leveraging non-vision sensors naturally integrated into daily wear, without adding burden to users, becomes essential. Motivated by this perspective, the next chapter investigates a gait estimation method using shoe-embedded IMUs, a minimal and unobtrusive sensor configuration well suited for continuous gait analysis in clinical and real-world environments.

## Chapter 5

# IMU-based Gait Estimation

Gait is a fundamental aspect of human mobility, and disruptions in normal gait can significantly reduce quality of life (QOL). Although recent advances in 3D gait analysis (3DGA) enable precise, quantitative assessments, these methods are typically confined to controlled laboratory environments and thus fail to accurately capture natural gait variability. Conversely, wearable IMU sensors offer cost-effective, portable solutions for capturing movements across diverse settings but face challenges such as invasiveness and sensor drift. In this chapter, we propose “Gait Inertial Poser (GIP),” a novel method estimating 3D full-body pose during straight walking on flat ground, using only two shoe-embedded IMU sensors. GIP initially estimates personalized body shapes from user attributes (height, weight, age, gender) and then employs a Transformer-based module to infer gait motion parameters from IMU data. To ensure temporal continuity and smoothness of the estimated motion, we further introduce a smoothing module based on a Variational Autoencoder (VAE), which further incorporates a specialized loss function that explicitly enforces kinematic constraints during foot-ground contact, thereby improving the overall estimation accuracy. Comprehensive experiments conducted on two public datasets quantitatively and qualitatively demonstrate that GIP achieves

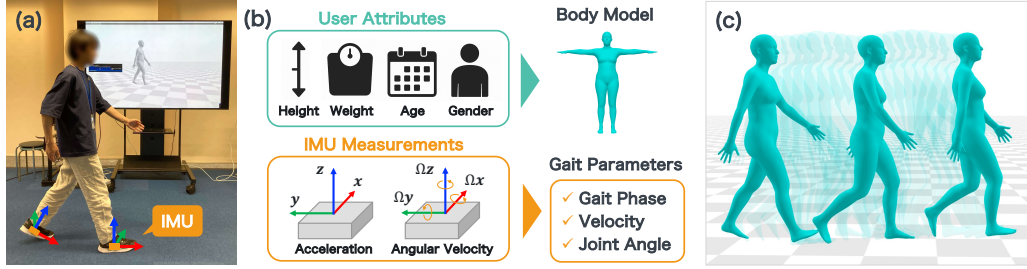


Figure 5.1: Overview of the proposed Gait Inertial Poser (GIP). (a) User wearing minimal inertial measurement units (IMUs) attached only to shoe soles, enabling unobtrusive daily gait measurement. (b) Input data consisting of easily obtainable user attributes (height, weight, age, gender) and IMU measurements (acceleration and angular velocity). These inputs allow estimation of the user’s personalized body model and gait parameters, including gait phase, velocity, and joint angles. (c) Final output visualizing smooth, accurate full-body walking motions tailored to the user’s characteristics.

high accuracy in straight-line walking. This approach overcomes limitations of traditional laboratory-based methods, opening new opportunities for real-time monitoring and remote rehabilitation in everyday environments.

## 5.1 Background

Gait is the most fundamental mode of human locomotion, encompassing critical information about a person’s health and physical capabilities, such as gait phase, stride length, and muscle strength. Quantitative analysis of this information, known as gait analysis, is widely utilized in medical fields for diagnosing patients, monitoring disease progression, and evaluating treatment and rehabilitation effectiveness [8, 143]. It also plays a significant role in sports science and rehabilitation research. Traditionally, clinical gait analysis has relied heavily on subjective evaluations through

visual observation by healthcare professionals and self-reports by patients. However, these methods are prone to variability among evaluators and human error. While Instrumented Gait Analysis (IGA), utilizing optical motion capture systems and force plates, provides objective and precise measurements [144], it is confined to laboratory settings and often fails to accurately reflect natural walking behaviors in daily life [13, 115, 119].

Recently, there has been growing interest in gait analysis methods employing Inertial Measurement Units (IMUs) to overcome these limitations [47, 112]. IMUs are compact, lightweight, cost-effective, and easily wearable, making them suitable for continuous gait measurement in various everyday environments. A conventional IMU-based method reconstructs foot trajectories by double integrating acceleration data [41, 68, 135]. Additionally, numerous machine learning-based approaches have been proposed to estimate lower-limb or full-body joint angles during gait from IMU data [66, 97, 98, 121]. More recently, methods have emerged capable of estimating not only joint angles but also full-body motion trajectories using only a few body-worn IMUs [59, 149, 156, 157]. However, these methods suffer from reduced estimation accuracy due to drift issues, including cumulative trajectory errors resulting from integration, and inaccuracies in global orientation information input into neural networks. These drift issues arise from the lack of absolute positioning in IMU sensors, as well as the inherent noise and measurement errors in IMU data.

To address these issues, we propose a novel method named “Gait Inertial Poser (GIP).” As illustrated in Figure 5.1, (a) GIP employs only two IMU sensors embedded in shoes, enabling non-invasive and highly accurate full-body gait motion estimation during straight walking on flat ground, suitable for unobtrusive measurements in daily life. (b) Additionally, the method leverages individual user attributes such as height, weight, age, and gender, enabling precise motion estimation tailored to each user’s body shape. Specifically, user-specific body models are esti-

mated from these attributes, and detailed gait parameters, including gait phases, walking velocity, and joint angles, are derived from IMU acceleration and angular velocity data via deep learning models. (c) By integrating these estimations, our approach enables the reconstruction of natural and precise full-body gait poses reflecting individual characteristics.

A key feature of our proposed method is intentionally limiting motion estimation to linear gait patterns. This design choice is critical to enabling highly accurate and practical gait motion estimation. It achieves this using only two shoe-embedded IMUs and requires no additional wearable devices or global positioning, making the approach realistic and widely applicable. Such a focus is well aligned with IGA, which primarily aims to quantify gait motion itself rather than absolute travel direction, and reflects typical clinical protocols (e.g., 10 m Walk Test) where straight and level walkways are standard. By restricting motion estimation to straight-line walking, we can avoid the drift issues typically observed in the conventional IMU-based full-body motion estimation methods. Unlike these methods, our approach does not require integrating angular velocities for global orientation or integrating accelerations to reconstruct displacement, and instead relies solely on IMU measurements within the sensor's local coordinate system, thereby avoiding integration-induced drift accumulation and enabling highly accurate gait motion estimation. Furthermore, we incorporate a Transformer-based deep learning model and a loss function inspired by Zero Velocity Update (ZUPT), achieving stable and accurate gait pattern estimation. We also employ a Variational Autoencoder (VAE) to enhance the temporal smoothness of the estimated motions.

To validate the effectiveness of our proposed method, we conducted quantitative and qualitative experiments using publicly available datasets such as the AIST Gait Database 2019 [64] and UnderPressure dataset [96]. Additionally, we developed a practical demonstration system utilizing commercially available IMU-equipped shoes to confirm the practical applica-

bility of our method in real-world environments. In particular, we adopted a consumer-grade IMU device, ORPHE CORE [136], embedded in sneakers designed specifically for this device. This setup allows consistent and vibration-resistant sensor placement inside the shoe sole, mitigating variability due to IMU mounting positions or shoe type.

The main contributions of this study are summarized as follows:

- We propose a non-invasive and highly accurate gait motion estimation method (Gait Inertial Poser) that naturally integrates into daily life using only two IMUs embedded in shoes.
- Our model comprises multiple modules that estimate personalized body models from user attributes and accurately predict gait parameters from IMU data. It also incorporates a VAE to ensure smooth motion reconstruction.
- The effectiveness and practicality of the proposed method are experimentally validated using publicly available datasets and a demonstration system with commercial IMU devices.

## 5.2 Method

We propose a novel approach, **Gait Inertial Poser (GIP)**, enabling non-invasive and accurate full-body motion estimation using a minimal number of IMU sensors naturally integrated into daily life. This method aims to enable gait analysis in diverse everyday environments beyond the constraints of laboratory settings. As illustrated in Figure 5.2, our framework consists primarily of three modules: the **Body Module**, the **Gait Module**, and the **Smoothing Module**. (a) First, subject-specific attributes, such as

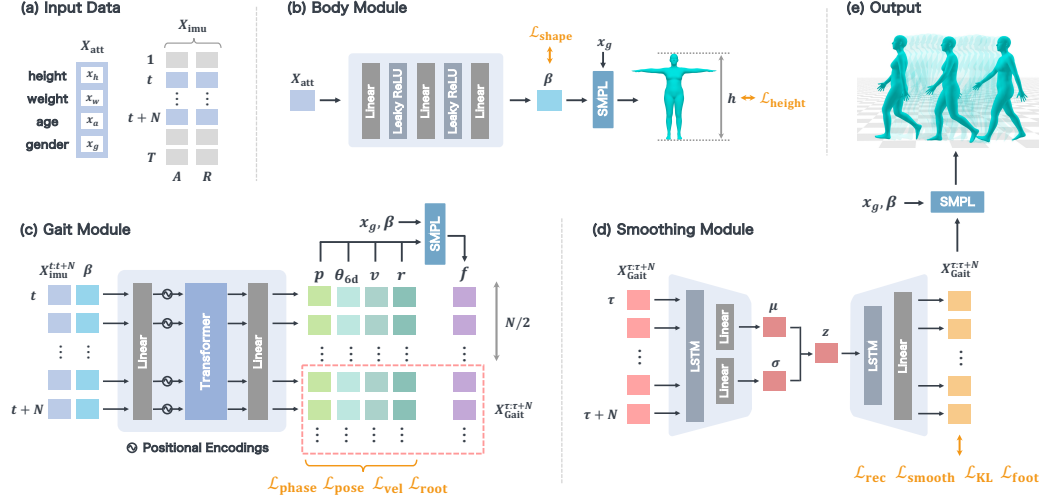


Figure 5.2: Detailed network architecture of the proposed method. (a) Input includes user attributes (height, weight, age, gender) and IMU measurements (acceleration, angular velocity). (b) The Body Module estimates personalized body shape parameters. (c) The Gait Module, based on a Transformer, simultaneously estimates gait phase, joint angles, velocities, root positions, and foot trajectories using IMU data. (d) The Smoothing Module employs a Variational Autoencoder (VAE) to enhance temporal smoothness and stability of the estimated motion, effectively reducing IMU drift through loss functions inspired by zero velocity updates (ZUPT). (e) Final output is a smooth, stable full-body walking motion reconstruction.

height, weight, age, and gender, along with acceleration and angular velocity data obtained from IMUs, serve as inputs. (b) The Body Module estimates personalized body shape parameters based on these individual attributes. (c) The Gait Module employs a Transformer-based deep learning model to simultaneously estimate multiple gait-related parameters from IMU data, including gait phase, joint angles, root joint velocity, root joint height, and foot height. (d) The Smoothing Module utilizes a Variational Autoencoder (VAE) to enhance the temporal smoothness and continuity of the estimated motion sequences. (e) Integrating these modules yields



natural and smooth full-body gait motions.

In the following subsections, we describe the sensor configuration used in the proposed method, detailed network architecture including each module, and the loss functions employed.

### 5.2.1 Sensor Configuration

IMUs are widely used for gait analysis and full-body MoCap due to their compatibility with human motion measurement, stemming from their wearability, low cost, high sampling rate, and real-time capabilities. An IMU consists of an accelerometer, gyroscope, and magnetometer, which together allow for tracking the velocity, position, and orientation of the body part to which the sensor is attached. The accelerometer measures the linear acceleration of an object along three axes, with a high capability to detect minute changes in acceleration. However, when acceleration data is double-integrated to determine position, noise and bias can accumulate, resulting in drift. The gyroscope measures the rotational velocity of an object along three axes and can respond immediately to dynamic movements to measure angular velocity, but it is prone to bias accumulation over time. The magnetometer detects the orientation of an object by measuring the Earth's magnetic field, but it is highly sensitive to magnetic disturbances (e.g., interference from electronic devices or metal), which can cause significant drift compared to the other two sensors. Therefore, in this study, we exclude magnetometer data and use the accelerometer and gyroscope values as inputs to the network, addressing their potential drift issues through network and loss function design.

Furthermore, sensor placement is also a critical factor in gait motion measurement. To capture natural gait patterns during everyday life, it is necessary to select a non-invasive sensor configuration that does not interfere with users' daily activities. The widely used commercial IMU sensor

system by Xsens [49] requires attaching 17 IMUs across the body for accurate full-body motion capture, making it unsuitable for long-term measurement in daily life or public settings. Recent advancements in machine learning-based pose estimation methods using IMU data have enabled diverse motion estimation with fewer IMUs, such as six sensors placed on the head, waist, hands, and legs [5, 46, 59, 137, 146, 155–157, 168], or just three sensors located on the wrist (smartwatch), pocket (smartphone), and head (earbuds) [95, 149]. In contrast, this study adopts an even more practical sensor configuration for daily use specifically for gait analysis, utilizing only two IMU sensors embedded in shoe soles. As smart insoles and smart shoes are becoming commercially available, embedding IMU sensors into regular footwear is increasingly feasible, enabling non-invasive, long-term monitoring of everyday movements without imposing additional user burden.

In this study, the acceleration data are denoted as  $\mathbf{A} = \{\mathbf{A}^t\}_{t=0}^T$ , and angular velocity data as  $\mathbf{R} = \{\mathbf{R}^t\}_{t=0}^T$ , where  $t$  represents the  $t$ -th frame in a sequence of length  $T$ . Given that our method uses two IMUs embedded in the shoe soles, the data at each frame are represented as  $\mathbf{A}^t \in \mathbb{R}^{2 \times 3}$  and  $\mathbf{R}^t \in \mathbb{R}^{2 \times 3}$ , respectively.

### 5.2.2 Network Details

The proposed method, **Gait Inertial Poser (GIP)**, is a deep learning-based framework designed for full-body motion estimation using a minimal number of IMU sensors. As illustrated in Figure 6.2, the network comprises three main modules: the **Body Module**, the **Gait Module**, and the **Smoothing Module**. The Body Module takes attribute data  $\mathbf{X}_{\text{att}}$  as input and estimates the body shape parameters  $\beta$  of a parametric human mesh model. The Gait Module uses IMU data  $\mathbf{X}_{\text{imu}}$  (acceleration  $\mathbf{A}$  and angular velocity  $\mathbf{R}$ ) and body shape parameters  $\beta$  to estimate multiple gait parameters

simultaneously, including the gait phase  $p$ , 6D joint angle representation  $\theta_{6d}$ , root joint velocity (walking speed)  $v$ , root joint height  $r$ , and foot height  $f$ . The Smoothing Module takes the gait parameters  $X_{\text{Gait}}^{\tau:\tau+N}$  estimated by the Gait Module using a sliding-window approach as input, incorporating overlapping frames from the subsequent window based on stride  $N/2$  to reduce discontinuities at window boundaries. Subsequently, a Variational Autoencoder (VAE) is employed to enhance the temporal smoothness of the estimated motion sequences. Details of each module are described below.

**Body Module.** The Body Module estimates body shape from simple attribute information without requiring a full-body scan. IMU data do not contain direct information about body shape; thus, IMU-based pose estimation methods typically use a common default body shape model for all subjects. However, differences in individual body shapes affect walking motions, making the use of personalized body models essential for accurate motion reconstruction. In particular, limb lengths influence stride length and IMU measurements. Accurately reproducing these individual differences is crucial for improving the precision of motion estimation.

To represent user-specific body shapes, we adopt the SMPL (Skinned Multi-Person Linear) model [79], a parametric human body model widely utilized in recent pose and shape estimation studies. This model effectively represents complex and diverse human poses and shapes using a limited number of parameters. The SMPL parameters consist of pose parameters  $\theta \in \mathbb{R}^{24 \times 3}$ , describing the relative rotations of 23 joints and the global rotation of the root joint, and shape parameters  $\beta \in \mathbb{R}^{10}$ , reflecting individual characteristics such as height, weight, and limb proportions. Using these parameters, the SMPL regression model estimates triangular mesh vertices  $M \in \mathbb{R}^{6890 \times 3}$  and 3D joint positions  $J \in \mathbb{R}^{24 \times 3}$ . Additionally, it incorporates the parameter  $d \in \mathbb{R}^3$  to capture the global translation of the person.

The Body Module estimates the body shape parameters  $\beta$  from subject attributes  $\mathbf{X}_{\text{att}} = \{x_h, x_w, x_a, x_g\}$  (height, weight, age, and gender) using a multi-layer perceptron (MLP). The SMPL model includes three variants: male, female, and neutral, and the appropriate variant is selected based on the gender attribute. The shape parameters  $\beta$  form a 10-dimensional vector comprising the top 10 principal components extracted via principal component analysis from a large-scale body scan dataset. By inputting these estimated body shape parameters along with IMU data into subsequent modules, our approach achieves highly accurate motion estimation that accounts for individual body differences.

**Gait Module.** The Gait Module simultaneously estimates multiple gait-related parameters such as gait phases, joint angles, walking speed, and foot trajectories from the body shape parameters and IMU sensor data. The network structure employs a Transformer-based deep learning model, effectively capturing temporal variations in IMU data for highly accurate motion estimation.

Human gait is characterized as periodic and repetitive movements of body segments, commonly described through “gait phases.” Incorporating gait phases facilitates a deeper understanding and analysis of the periodic gait mechanism. The gait cycle primarily consists of two phases: the stance phase, which accounts for approximately 62% of the cycle when the foot contacts the ground, and the swing phase, comprising about 38% when the foot is off the ground. For detailed analysis, the stance phase can be further divided into initial contact, loading response, mid-stance, terminal stance, and pre-swing phases. Similarly, the swing phase can be subdivided into initial swing, mid-swing, and terminal swing. This study specifically focuses on four key phases segmented by the following gait events: heel contact, toe contact, heel-off, and toe-off. The Gait Module identifies these gait phases using the input IMU sequential data and out-

puts the corresponding gait phase logits  $\mathbf{p} \in \mathbb{R}^4$ . The final gait phase is determined by selecting the maximum value from these logits.

In addition to gait phases, accurate gait motion estimation requires consideration of kinematic parameters such as joint movements, walking speed, and foot trajectories. Therefore, the module leverages the body shape parameter  $\beta$  estimated by the Body Module to simultaneously predict gait phase  $\mathbf{p}$ , 6-dimensional joint angle representation  $\theta_{6d}$ , root joint velocity  $\mathbf{v}$ , and root joint height  $r$ , based on the SMPL human body model. The 6D representation ( $\theta_{6d}$ ) is a continuous and rotation-invariant representation of joint angles [174], widely used in pose estimation tasks to mitigate issues like discontinuity and ambiguity inherent in other representations such as Euler angles or axis-angle. Parameters such as walking speed and stride length are particularly influenced by subject-specific body attributes like limb length, thus improving estimation accuracy through incorporating individual body shape parameters. Furthermore, the Transformer architecture is specifically designed to extract temporal features, ensuring stable temporal estimations of these parameters, inspired by prior studies [59,70] that adopted similar architectures for pose estimation from IMU data. Subsequently, by solving forward kinematics (FK) using a pre-trained SMPL joint estimation model with the predicted joint angles and body shape parameters, the module obtains full-body joint positions and derives foot trajectories  $\mathbf{f}$ . The estimated gait parameters are then input into the subsequent Smoothing Module for further motion refinement.

**Smoothing Module.** The Smoothing Module is designed to reduce temporal discontinuities in the motion parameter sequences estimated by the Gait Module, resulting in smooth and natural motion estimation. Because the Gait Module performs motion estimation using a sliding-window approach, discontinuities may arise at the boundaries between estimation windows. To address this issue, the Smoothing Module processes over-

lapping frames between adjacent windows by using a stride  $N/2$ , enabling continuous correction of motion near these boundaries.

Specifically, when utilizing the gait parameter sequence  $\mathbf{X}_{\text{Gait}}^{\tau:\tau+N}$  estimated by the Gait Module, frames from the latter half of each estimation window overlap with frames from the first half of the subsequent window, thereby seamlessly connecting otherwise discontinuous estimations. Moreover, we introduce a loss function inspired by the classical Zero Velocity Update (ZUPT) algorithm, a widely used technique in IMU-based gait analysis that reduces integration-induced positional drift by resetting sensor velocity to zero during foot-ground contact. By leveraging the gait phase information estimated by the Gait Module, our proposed loss function explicitly imposes kinematic constraints on foot velocity and height during identified foot-ground contact periods, thereby effectively mitigating IMU drift and improving motion estimation accuracy. The details of this loss function are described in subsequent sections.

The module employs a network architecture based on a Variational Autoencoder (VAE). Sequential input data are first processed by an LSTM to estimate the mean  $\mu$  and standard deviation  $\sigma$  of latent variables, from which a latent representation  $\mathbf{z}$  is sampled. This latent representation is then fed into a decoder to reconstruct a continuous and smooth motion sequence. This design ensures temporal stability in the estimated motions, resulting in smooth gait patterns well-suited for practical applications in real-world environments.

### 5.2.3 Loss Functions

Our network architecture allows each module to be trained separately, ensuring stable learning without mutual interference and facilitating efficient convergence. Below, we detail the loss functions employed for individual module training.

**Body Module.** The loss function  $\mathcal{L}_{\text{body}}$  used for training the Body Module consists of two terms: a shape loss  $\mathcal{L}_{\text{shape}}$  and a height loss  $\mathcal{L}_{\text{height}}$ , as described in Equation 5.1. Here,  $\lambda_{\text{shape}}$  and  $\lambda_{\text{height}}$  are weighting coefficients used as hyperparameters to balance each loss component effectively.

$$\mathcal{L}_{\text{body}} = \lambda_{\text{shape}}\mathcal{L}_{\text{shape}} + \lambda_{\text{height}}\mathcal{L}_{\text{height}} \quad (5.1)$$

$$\mathcal{L}_{\text{shape}} = |\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}|_1 \quad (5.2)$$

$$\mathcal{L}_{\text{height}} = |\mathbf{h} - \hat{\mathbf{h}}|_2^2 \quad (5.3)$$

The shape loss  $\mathcal{L}_{\text{shape}}$  calculates the mean absolute error (MAE) between the estimated shape parameters  $\boldsymbol{\beta}$  and the ground truth  $\hat{\boldsymbol{\beta}}$ . Since each dimension of  $\boldsymbol{\beta}$  corresponds to principal components obtained via PCA on body shape data, dimensions may have different variance scales. Therefore, using MAE ensures stable training by preventing disproportionately large contributions from dimensions with higher variance.

Although  $\mathcal{L}_{\text{shape}}$  effectively optimizes multiple aspects such as limb length and body thickness, height is particularly crucial in gait analysis due to its significant influence on stride length and joint angle estimations. Thus, we introduce an additional height-specific loss term,  $\mathcal{L}_{\text{height}}$ , which explicitly guides the model to more accurately predict body height. Here, body height is defined as the distance from the soles of the feet to the top of the head when applying the estimated and ground truth shape parameters ( $\boldsymbol{\beta}$  and  $\hat{\boldsymbol{\beta}}$ ) to the SMPL model in the default T-pose. This loss term calculates the mean squared error (MSE) between the predicted height  $\mathbf{h}$  and the ground truth height  $\hat{\mathbf{h}}$ .

In practice, the height loss is computed by generating a body mesh using the estimated  $\boldsymbol{\beta}$  parameters with gender-specific SMPL models (male or female) and measuring the vertical distance between the vertex at the top of the head and the midpoint of the left and right heel vertices. This calculated height is then compared with the ground truth to compute the

height loss. This formulation ensures accurate and stable estimation of individual-specific body shapes, contributing to precise gait analysis.

**Gait Module.** The loss function  $\mathcal{L}_{\text{gait}}$  for training the Gait Module combines multiple components, each measuring the accuracy of different gait-related parameters:

$$\begin{aligned} \mathcal{L}_{\text{gait}} = & \lambda_{\text{phase}}\mathcal{L}_{\text{phase}} + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}} + \lambda_{\text{ori}}\mathcal{L}_{\text{ori}} \\ & + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{root}}\mathcal{L}_{\text{root}}, \end{aligned} \quad (5.4)$$

where each  $\lambda$  represents a hyperparameter balancing the contribution of each component.

The gait phase loss  $\mathcal{L}_{\text{phase}}$  evaluates the accuracy of gait phase estimation through cross-entropy loss for a multi-class classification task, defined as follows:

$$\mathcal{L}_{\text{phase}} = -\frac{1}{2T} \sum_{t=1}^T \left( \log p_{\text{left},t}^{(\hat{y}_{\text{left},t})} + \log p_{\text{right},t}^{(\hat{y}_{\text{right},t})} \right), \quad (5.5)$$

where  $p_{\text{left},t}^{(c)}$  and  $p_{\text{right},t}^{(c)}$  represent the predicted probabilities for class  $c$  of the left and right feet, respectively, at frame  $t$ . The ground truth labels for the left and right feet at frame  $t$  are denoted as  $\hat{y}_{\text{left},t}$  and  $\hat{y}_{\text{right},t}$ , respectively, each indicating one of the classes.  $T$  denotes the total number of frames.

The remaining losses (pose, orientation, velocity, and root height) are computed using mean squared error (MSE) between predicted and ground



truth values:

$$\mathcal{L}_{\text{pose}} = \frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K \|\theta_t^k - \hat{\theta}_t^k\|_2^2, \quad (5.6)$$

$$\mathcal{L}_{\text{ori}} = \frac{1}{T} \sum_{t=1}^T \|\theta_t^0 - \hat{\theta}_t^0\|_2^2, \quad (5.7)$$

$$\mathcal{L}_{\text{vel}} = \frac{1}{T} \sum_{t=1}^T \|v_t - \hat{v}_t\|_2^2, \quad (5.8)$$

$$\mathcal{L}_{\text{root}} = \frac{1}{T} \sum_{t=1}^T \|r_t^{(z)} - \hat{r}_t^{(z)}\|_2^2, \quad (5.9)$$

where  $K$  is the number of joints,  $\theta_t^k$  represents the estimated 6-dimensional joint angle representation for the  $k$ -th joint at frame  $t$ ,  $\theta_t^0$  is the estimated 6-dimensional orientation of the root joint at frame  $t$ ,  $v_t$  denotes the estimated root joint velocity (walking speed) at frame  $t$ , and  $r_t^{(z)}$  indicates the estimated root joint height at frame  $t$ . Corresponding ground truth values are denoted with hats ( $\hat{\cdot}$ ). By integrating these multiple loss components, the Gait Module accurately estimates the diverse and crucial parameters essential for detailed gait analysis.

**Smoothing Module.** The loss function for the Smoothing Module,  $\mathcal{L}_{\text{smooth}}$ , is designed based on a Variational Autoencoder (VAE) framework and comprises multiple components. These components include reconstruction loss, smoothing loss, translation loss, foot-ground contact constraints, and the KL divergence term. Specifically, it is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{smooth}} = & \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{acc}} \mathcal{L}_{\text{acc}} \\ & + \lambda_{\text{fpos}} \mathcal{L}_{\text{fpos}} + \lambda_{\text{fvel}} \mathcal{L}_{\text{fvel}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \end{aligned} \quad (5.10)$$

where each  $\lambda$  represents a weighting hyperparameter to balance the contribution of each term.

The reconstruction loss  $\mathcal{L}_{\text{rec}}$  is calculated as the Mean Squared Error (MSE) between the predicted gait parameters (joint angles, orientations, velocities, root joint height, foot joint height) and their ground truth values, ensuring accurate reproduction of the input motion sequences:

$$\mathcal{L}_{\text{rec}} = \frac{1}{TD} \sum_{t=1}^T \sum_{d=1}^D \|z_{t,d} - \hat{z}_{t,d}\|_2^2, \quad (5.11)$$

where  $z_{t,d}$  and  $\hat{z}_{t,d}$  denote the predicted and ground truth motion parameters at frame  $t$  and dimension  $d$ , respectively.  $D$  represents the total number of gait parameter dimensions.

The acceleration loss  $\mathcal{L}_{\text{acc}}$  is defined by minimizing the second-order temporal derivative of the predicted motion parameters. This encourages temporal smoothness and continuity in motion estimations:

$$\mathcal{L}_{\text{acc}} = \frac{1}{TD} \sum_{t=2}^{T-1} \sum_{d=1}^D \|\ddot{x}_{t,d}\|_2^2, \quad (5.12)$$

where the second derivative term  $\ddot{x}_t$  is approximated via finite differences as follows:

$$\ddot{x}_t \approx \frac{x_{t+1} - 2x_t + x_{t-1}}{\Delta t^2}. \quad (5.13)$$

Here,  $x_t \in \mathbb{R}^{K \times 3}$  denotes 3D joint positions at frame  $t$ , which are obtained via the forward kinematics function  $\text{FK}(\beta, \theta_t, r_t)$  using the pre-trained SMPL joint regressor.

Moreover, inspired by the classical Zero Velocity Update (ZUPT) algorithm, we introduce constraints during periods identified as foot-ground contact phases, leveraging gait phase information  $p$  from the Gait Module. Specifically, we define two loss functions: a foot height loss,  $\mathcal{L}_{\text{fpos}}$ , and a foot velocity loss,  $\mathcal{L}_{\text{fvel}}$ .

The foot height loss  $\mathcal{L}_{\text{fpos}}$  minimizes the vertical displacement of foot

sole vertices, promoting accurate foot-ground contact constraints:

$$\mathcal{L}_{\text{fpos}} = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} \|f_t^{(z)}\|_2^2, \quad (5.14)$$

where  $f_t^{(z)}$  is the predicted vertical position of the foot sole at frame  $t$ , and  $\mathcal{C}$  is the set of foot-ground contact frames and vertices identified by gait phase predictions.

Similarly, the foot velocity loss  $\mathcal{L}_{\text{fvel}}$  suppresses horizontal and vertical foot movements during identified ground-contact periods:

$$\mathcal{L}_{\text{fvel}} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \|\dot{f}_i\|_2^2, \quad (5.15)$$

where  $\dot{f}_i$  denotes the velocity of foot sole at frame  $t$ .

Lastly, the KL divergence term  $\mathcal{L}_{\text{KL}}$  regularizes the latent variable distribution to approximate a standard normal distribution, following the standard VAE objective:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum_{l=1}^L \left( 1 + \log \sigma_l^2 - \mu_l^2 - \sigma_l^2 \right), \quad (5.16)$$

where  $L$  is the dimensionality of the latent space, and  $\mu_l, \sigma_l^2$  represent the mean and variance of the latent distribution, respectively.

This combined loss function ensures accurate reconstruction, temporal smoothness, robust velocity integration, effective drift suppression during foot contact, and stable latent representations, thereby significantly enhancing the accuracy and smoothness of gait motion estimation.

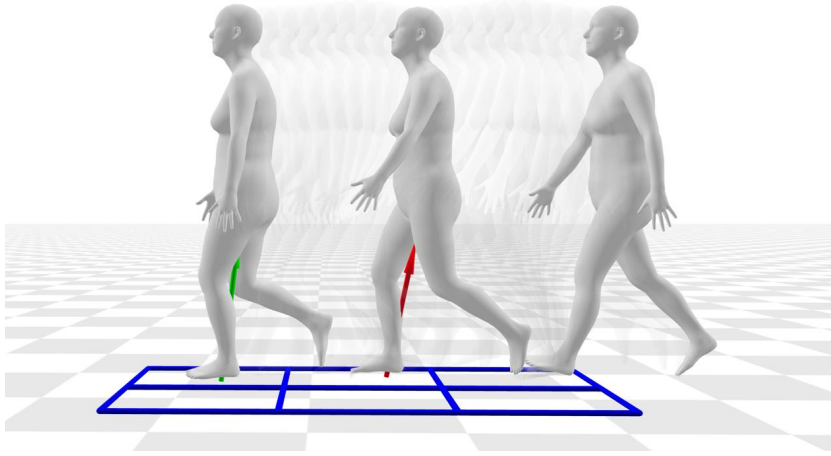


Figure 5.3: Example of the gait motion data from the AIST Gait Database. Subjects walked naturally along a straight path, stepping on force plates (indicated in blue), allowing simultaneous capture of motion and ground reaction force (GRF) data (indicated in red and green). The GRF data were used to define gait phases such as heel-strike and toe-off.

## 5.3 Experiments

### 5.3.1 Datasets

Our proposed approach is a data-driven motion estimation framework utilizing machine learning models. Thus, in this study, we used the following two existing gait datasets for model training and evaluation.

The first dataset is the “AIST Gait Database 2019” [64], published by the National Institute of Advanced Industrial Science and Technology (AIST). This database comprises gait motion data collected according to protocols approved by the AIST Human Ergonomics Experiment Committee. After excluding data with confirmed measurement errors, the dataset consists of 3,424 walking trials conducted by a total of 352 healthy adults, including 1,609 trials from 167 males and 1,815 trials from 185 females. The par-

ticipants span a wide demographic range, with ages from 20 to 78 years (mean  $51.1 \pm 18.7$ ), heights from 138 to 185 cm (mean  $162.9 \pm 8.4$ ), and body weights from 34 to 100 kg (mean  $59.5 \pm 10.2$ ). The data were captured at 200 FPS, amounting to a total of 1 million frames. Measurements were performed using a Vicon optical MoCap system, with subjects instructed to walk naturally along a straight line at their preferred walking speed. Additionally, the dataset includes ground reaction force (GRF) data captured using force plates. During data collection, participants walked over force plates located near the center of the laboratory setup, allowing simultaneous recording of motion data and GRF data (Figure 5.3).

Gait phase labels used for training our method were derived using this GRF data. The gait cycle was segmented into four phases based on gait events: “heel-strike,” “toe-contact,” “heel-off,” and “toe-off.” Heel-strike and toe-off timings were defined based on the initiation and cessation of ground reaction forces, respectively. In contrast, toe-contact and heel-off phases were labeled when the velocity of specific mesh vertices on the foot soles fell below a predefined threshold. This dataset is highly suitable for evaluating gait estimation methods due to the diversity of its subjects. However, because the dataset relies on force plates, each trial contains only a very short sequence of a few seconds and does not include actual IMU sensor measurements. To address these limitations, we supplemented the evaluation using synthesized IMU data (described later) and also utilized another dataset, which contains longer gait sequences and actual IMU measurement data.

The second dataset used is the “UnderPressure dataset” [96], originally created to address the foot-skating issue common in computer graphics-based motion estimation tasks. This dataset contains motion data recorded from 10 participants wearing insoles equipped with IMU sensors and foot-pressure sensors. The participants consist of 8 males and 2 females, aged 21–55 years (mean  $31.4 \pm 11.7$ ), with heights ranging from 167 to 184 cm

(mean  $176.4 \pm 7.7$  cm) and weights from 65 to 91 kg (mean  $77.9 \pm 9.3$  kg). Each participant performed motion sequences with an average duration of about 1.5 minutes. It includes diverse movements such as walking at slow, medium, and fast speeds, walking with random directional changes, interacting with objects, and stair climbing. Since our experiments focus on straight-line gait motions, we excluded sequences involving interactions with objects and extracted segments where the yaw angular velocity of the root joint remained below a specified threshold. The total duration of these extracted segments amounted to 170K frames (28.5 minutes at 100 FPS). For gait phase labeling, we defined the four phases mentioned above based on pressure detection using the 16 foot-pressure sensors per foot, divided into front and rear sections.

### 5.3.2 Data Synthesis

Creating extensive motion capture datasets requires significant effort. Consequently, many IMU-based approaches typically employ synthesized IMU data in addition to real data for training and evaluation. Following this common practice, we generated synthetic acceleration and angular velocity data for cases in the AIST Gait Database and UnderPressure datasets where actual IMU data was unavailable.

Accurate synthesis of IMU data requires reproducing not only skeletal poses but also individual-specific body shapes. Therefore, we generate mesh models as pseudo-ground-truth data to represent the actual body shape of each subject. For the AIST Gait Database, we reproduced individual body shapes using “Mosh++” [80, 90], an optimization method that fits the SMPL human mesh model to optical marker trajectory data. Prior to optimization, we manually established correspondences between optical marker positions and SMPL mesh vertices to accurately generate subject-specific mesh models. In contrast, the UnderPressure dataset con-

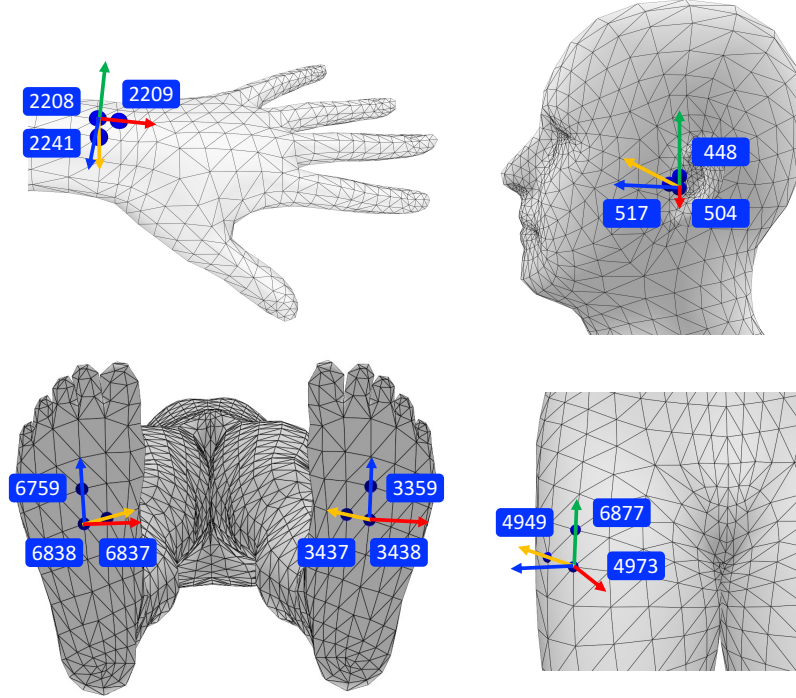


Figure 5.4: Synthetic IMU sensor positions defined on SMPL mesh vertices (blue labels indicate SMPL vertex IDs). The foot soles correspond to the proposed method, while positions on the left wrist, left ear, and right thigh simulate IMU signals for baseline methods (smartwatch, earphone, smartphone in pocket). The yellow arrows indicate vectors used to define the mesh face normal for establishing IMU coordinate systems.

tains motion data recorded using commercial high-performance IMU sensors (Xsens) without optical markers. Therefore, we fitted the SMPL model by solving an inverse kinematics (IK) algorithm based on skeletal landmarks common to both the Xsens model and the SMPL model.

Next, we synthesized IMU data from the obtained SMPL mesh models. Specifically, to evaluate our proposed method and baseline methods described later, we created synthetic IMU data for five locations: the left wrist, left ear, right pocket, and both foot soles, as shown in Figure 5.4. The IMU coordinate systems were defined based on selected triangular mesh

faces by using the vectors of triangle edges, the normal vector to the triangle face, and their cross product. Using this IMU coordinate system, we computed the orientation and acceleration in the global coordinate system during motions and subsequently transformed these data into synthetic acceleration and angular velocity data in the IMU sensor coordinate system.

### 5.3.3 Implementation Details

We implemented our proposed and baseline methods using PyTorch as the deep learning framework. Model training and evaluation were conducted on a Linux server equipped with an NVIDIA RTX A6000 GPU. Specifically, the Body Module consisted of a three-layer multi-layer perceptron (MLP) with hidden layers of 128, 256, and 128 units, respectively. The Gait Module employed a Transformer-based encoder with 8 attention heads and 4 encoder layers, a latent dimension of 256, and an input window size of 50 frames. The Smoothing Module was implemented as a Variational Autoencoder (VAE) with a latent dimension of 64 and an LSTM dimension of 256.

For training, we employed the Adam optimizer [63] with an initial learning rate of  $1 \times 10^{-3}$ , trained for 500 epochs. The dataset was divided into training, validation, and test subsets with a ratio of 8:1:1, and a mini-batch size of 32 was used. Model training and testing were conducted at 200 fps for the AIST Gait Database and at 100 fps for the UnderPressure dataset. Loss function weights were empirically optimized through preliminary experiments and set as follows:

$$\begin{aligned} (\lambda_{\text{shape}}, \lambda_{\text{height}}) &= (0.1, 100), \\ (\lambda_{\text{phase}}, \lambda_{\text{pose}}, \lambda_{\text{ori}}, \lambda_{\text{vel}}, \lambda_{\text{root}}) &= (1, 100, 100, 10, 10), \\ (\lambda_{\text{rec}}, \lambda_{\text{smooth}}, \lambda_{\text{fpos}}, \lambda_{\text{fvel}}, \lambda_{\text{KL}}) &= (1000, 1 \times 10^5, 1, 0.1). \end{aligned}$$



To identify foot-ground contact phases in the AIST Gait Database, we set the velocity threshold of the toes and heels on the foot sole to 0.1 m/s. For the UnderPressure dataset, straight-walking segments were extracted by analyzing the root joint velocity direction. The method compares the mean horizontal heading over 15-past and future windows and labels frames as non-straight when the heading change exceeds  $2^\circ$ , also excluding a margin of neighboring 10 frames, all parameter values were empirically determined. During inference, the root joint trajectory was calculated by integrating the estimated root joint velocity, starting from the origin on the horizontal plane, combined with the estimated root joint height.

Finally, following Zhang et al. [59], we applied low-pass filtering to both synthetic and real IMU data to reduce differences in waveform characteristics. Specifically, a 4th-order Butterworth low-pass filter with a cutoff frequency of 10 Hz and a sampling frequency of 100 Hz was employed.

### 5.3.4 Baseline Methods

To evaluate the performance of our proposed method, we employed the following four baseline methods for comparison:

**Integration.** The simplest baseline method, which estimates trajectories by directly integrating acceleration and angular velocity data from shoe-embedded IMU sensors. The initial position and orientation are assumed to be known. However, it is widely recognized that this method suffers from cumulative integral drift, significantly degrading accuracy over time.

**Integration + ZUPT.** An enhanced integration method that incorporates Zero-Velocity Update (ZUPT) to mitigate drift. Using gait-phase information estimated by our proposed method, foot-ground contact periods are identified. In addition to setting velocities to zero during ground contacts,

the method also removes accumulated velocity drift between successive ground contacts. Specifically, drift correction is applied by linearly interpolating velocities between the start of each ground contact and the end of the next, subtracting the interpolated drift from the original velocity data. Furthermore, positional drift along the vertical axis is corrected by adjusting trajectories so that the foot positions align with the ground plane at the beginning and end of each swing phase.

**IMUPoser [95].** A pose estimation method utilizing consumer-grade wearable devices such as smartphones, smartwatches, and earbuds. In this experiment, IMU data from three locations (left wrist, right pocket, and left ear) were used. While IMUPoser accurately estimates root-relative body pose and segment movements, it does not estimate the absolute position (translation) of the root joint. Following prior work [5], we employed the Versatile Quaternion-based Filter (VQF), a high-precision orientation estimation method that accounts for gravity, to transform 6-DoF IMU data (acceleration and angular velocity) into orientation and acceleration in the global coordinate system.

**MobilePoser [149].** An extension of IMUPoser [95], employing the same IMU sensor configuration but additionally estimating the absolute position (translation) of the root joint. Thus, MobilePoser estimates both pose and overall body translation, enhancing its applicability to a broader range of environments. As with IMUPoser, we utilized VQF to convert IMU data into the global coordinate frame.

### 5.3.5 Evaluation Metrics

We employed the following metrics to evaluate the performance of the proposed method and baseline methods:

**MPJPE, PEL-MPJPE.** These metrics evaluate the accuracy of pose estimation. Mean per joint position error (MPJPE) calculates the average Euclidean distance between the estimated and ground truth (GT) joint positions in global coordinates over all joints and frames. The Pelvis-aligned MPJPE (PEL-MPJPE) evaluates joint errors after aligning the root joint positions of predictions and GT. The unit is centimeters (cm), computed as:

$$\text{MPJPE} = \frac{1}{TK} \sum_{t=1}^T \sum_{k=1}^K \|J_{t,k} - \hat{J}_{t,k}\|_2 \quad (5.17)$$

where  $T$  is the number of frames,  $K$  is the number of joints, and  $J_{t,k}$  and  $\hat{J}_{t,k}$  are the predicted and GT joint coordinates for frame  $t$  and joint  $k$ .

**PVE-P, PVE-T.** These metrics assess SMPL mesh vertex prediction accuracy. They calculate the per vertex error (PVE), averaging the Euclidean distances between estimated and GT mesh vertices across all vertices and frames. PVE-P evaluates errors with aligned root coordinates, thus incorporating pose information (“P” for Pose). Since PVE-P combines shape and pose errors, PVE-T evaluates the pure shape estimation accuracy by measuring errors at the default T-pose. Units are centimeters (cm), calculated as:

$$\text{PVE} = \frac{1}{TQ} \sum_{t=1}^T \sum_{q=1}^Q \|M_{t,v} - \hat{M}_{t,v}\|_2 \quad (5.18)$$

where  $Q$  is the number of vertices, and  $M_{t,q}$  and  $\hat{M}_{t,q}$  represent the predicted and GT vertex coordinates for frame  $t$  and vertex  $q$ .

**Joint Angle.** This metric measures joint angle estimation accuracy. Although SMPL uses axis-angle representation, angles are converted to Euler angles (ZYX order) for evaluation. Angle differences between predicted and GT values are computed for each axis, joint, and frame, and averaged.

Units are degrees ( $^\circ$ ):

$$\text{Angle Error} = \frac{1}{3TK} \sum_{t=1}^T \sum_{k=1}^K \sum_{a \in \{x,y,z\}} |\theta_{t,k,a} - \hat{\theta}_{t,k,a}| \quad (5.19)$$

where  $\theta_{t,k,a}$  and  $\hat{\theta}_{t,k,a}$  denote the predicted and GT joint angles for frame  $t$ , joint  $k$ , and axis  $a$ .

**InterFoot.** This metric evaluates stride length accuracy by measuring the difference between the predicted and GT distances between feet, averaged over all frames. Units are centimeters (cm):

$$\text{InterFoot Error} = \frac{1}{T} \sum_{t=1}^T \left| \|f_t^L - f_t^R\|_2 - \|\hat{f}_t^L - \hat{f}_t^R\|_2 \right|, \quad (5.20)$$

where  $f_t^L, f_t^R$  represent the predicted positions of the left and right feet at frame  $t$ , and  $\hat{f}_t^L, \hat{f}_t^R$  are the corresponding ground-truth positions.

**FootTraj.** This metric evaluates the accuracy of foot trajectory estimation by measuring the Euclidean distance between the predicted and ground truth 3D foot positions for both left and right feet. The result is averaged over all frames and both feet. Units are centimeters (cm):

$$\text{FootTraj Error} = \frac{1}{2T} \sum_{t=1}^T \sum_{s \in \{\text{left}, \text{right}\}} \|f_{t,s} - \hat{f}_{t,s}\|_2 \quad (5.21)$$

where  $f_{t,s}$  and  $\hat{f}_{t,s}$  represent the predicted and ground truth positions of foot  $s$  (left or right) at frame  $t$ .

**FootVel.** This metric evaluates the accuracy of foot velocity by comparing the temporal derivatives of the predicted and ground truth foot trajectories. The velocity is computed using finite differences, and the error is defined as:

$$\text{FootVel Error} = \frac{1}{2(T-1)} \sum_{t=1}^{T-1} \sum_{s \in \{\text{left}, \text{right}\}} \|\dot{f}_{t,s} - \dot{\hat{f}}_{t,s}\|_2, \quad (5.22)$$

where  $\dot{f}_{t,s}$  denotes the velocity of foot  $s$  (left or right) at frame  $t$ , approximated as  $\dot{f}_{t,s} \approx (f_{t+1,s} - f_{t,s}) / \Delta t$ .

**RootVel.** This metric evaluates the accuracy of translational velocity of the root joint. Similar to FootVel, velocity is computed using finite differences, and the error is defined as:

$$\text{RootVel Error} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|\dot{r}_t - \hat{\dot{r}}_t\|_2, \quad (5.23)$$

where  $\dot{r}_t$  denotes the root joint velocity at frame  $t$ , approximated by  $\dot{r}_t \approx (r_{t+1} - r_t) / \Delta t$ .

**Phase.** This metric evaluates the accuracy of gait phase classification using the macro-averaged F1 score across the four gait phases. At each frame, the predicted gait phase label is compared with the ground truth label, and the F1 score is computed for each class individually across all frames. The final metric is the macro-average over all classes:

$$\text{Macro-F1} = \frac{1}{C} \sum_{c=1}^C \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}, \quad (5.24)$$

where  $C = 4$  is the number of gait phase classes, and  $\text{Precision}_c$  and  $\text{Recall}_c$  represent the precision and recall for class  $c$ , calculated across the entire sequence of frames.

### 5.3.6 Comparison with Baseline Methods

In this section, we present quantitative and qualitative evaluation results from comparative experiments conducted using the AIST Gait Database and the UnderPressure dataset to compare our proposed method against baseline methods.

Table 5.1: Performance comparison of foot trajectory and pose estimation methods across different datasets (AIST Gait Database and UnderPressure Dataset) and evaluation metrics. Bold values indicate the highest accuracy achieved by each metric within each dataset.

| Method                       | MPJPE<br>[cm] (↓) | PEL-MPJPE<br>[cm] (↓) | PVE-P<br>[cm] (↓) | PVE-T<br>[cm] (↓) | JointAngle<br>[degree] (↓) | InterFoot<br>[cm] (↓) | FootTraj<br>[cm] (↓) | FootVel<br>[cm/s] (↓) | RootVel<br>[cm/s] (↑) | Phase<br>[F1 Score] (↑) |
|------------------------------|-------------------|-----------------------|-------------------|-------------------|----------------------------|-----------------------|----------------------|-----------------------|-----------------------|-------------------------|
| <b>AIST Gait Database</b>    |                   |                       |                   |                   |                            |                       |                      |                       |                       |                         |
| Integration                  | -                 | -                     | -                 | -                 | -                          | 8.25                  | 11.44                | 18.63                 | -                     | -                       |
| Integration + ZUPT           | -                 | -                     | -                 | -                 | -                          | 7.26                  | 8.04                 | <b>17.93</b>          | -                     | <b>0.96</b>             |
| IMUPoser (3IMUs)             | -                 | 3.06                  | 3.69              | 4.40              | 2.69                       | 5.30                  | -                    | -                     | -                     | -                       |
| MobilePoser (3IMUs)          | 9.11              | <b>2.90</b>           | <b>3.60</b>       | 4.40              | 2.74                       | 4.12                  | 9.59                 | 33.07                 | 19.19                 | -                       |
| Ours (2IMUs)                 | <b>5.73</b>       | 3.18                  | 4.36              | <b>1.58</b>       | <b>2.26</b>                | <b>3.11</b>           | <b>5.32</b>          | 22.84                 | <b>8.25</b>           | <b>0.96</b>             |
| <b>UnderPressure Dataset</b> |                   |                       |                   |                   |                            |                       |                      |                       |                       |                         |
| Integration                  | -                 | -                     | -                 | -                 | -                          | 24.47                 | 28.20                | 36.15                 | -                     | -                       |
| Integration + ZUPT           | -                 | -                     | -                 | -                 | -                          | 14.83                 | 19.91                | 37.23                 | -                     | <b>0.89</b>             |
| IMUPoser (3IMUs)             | -                 | 4.26                  | 4.90              | 4.37              | 3.10                       | 8.08                  | -                    | -                     | -                     | -                       |
| MobilePoser (3IMUs)          | 28.83             | 4.14                  | <b>4.71</b>       | 4.37              | 3.12                       | 6.99                  | 31.74                | 44.58                 | 26.89                 | -                       |
| Ours (2IMUs)                 | <b>10.51</b>      | <b>3.87</b>           | 4.84              | <b>2.06</b>       | <b>2.29</b>                | <b>4.25</b>           | <b>12.35</b>         | <b>26.70</b>          | <b>8.45</b>           | <b>0.89</b>             |

**Quantitative Evaluation.** Table 5.1 summarizes the quantitative evaluation results. Firstly, our proposed method exhibited superior performance across both datasets for MPJPE, PVE-T, and JointAngle metrics, demonstrating its effectiveness in accurately estimating absolute joint positions and reconstructing body shapes. The improved MPJPE accuracy compared to MobilePoser is primarily attributed to two factors. First, the estimation accuracy of the root joint velocity (RootVel) during straight walking significantly affects the accuracy of joint positions. Our proposed method estimates velocity more accurately than MobilePoser, resulting in root joint positions that are closer to the ground truth (GT) and consequently reducing the relative joint position error (MPJPE). Second, MobilePoser utilizes IMU data in the global coordinate system, potentially causing drift errors during extended measurements. Consumer-grade devices often include 6DoF IMUs without magnetometers, making global orientation estimation challenging and susceptible to drift over time. Following previous research [5], we applied the Versatile Quaternion-based Filter (VQF)

for global coordinate transformations. However, VQF accumulates errors for data recordings longer than several tens of seconds. In contrast, our method utilizes IMU data directly in the sensor coordinate system, thereby significantly mitigating drift issues and maintaining stable accuracy during straight walking motions.

Conversely, our method slightly underperformed IMUPoser and MobilePoser in the PEL-MPJPE and PVE-P metrics. This minor discrepancy likely arises because IMUPoser employs more IMUs (three sensors) compared to our two sensors. However, the observed differences were minimal, on the order of a few millimeters. Furthermore, the strong performance in PVE-T indicates the efficacy of our method's personalized body shape reconstruction approach, considering individual attributes such as height and weight. Regarding JointAngle evaluation, our method achieved the best performance, although the differences among all methods were minor. This suggests that even with fewer IMUs, accurate pose estimation is feasible for simple motions like walking.

Additionally, our method demonstrated high accuracy across both datasets in metrics like InterFoot (inter-foot distance) and FootTraj (foot trajectory), highlighting its ability to reconstruct personalized body shapes and effectively suppress drift. Regarding FootVel (foot velocity), Integration+ZUPT performed best with the synthetic IMU data of the AIST Gait Database, whereas our method was superior with the real, noisy IMU data from the UnderPressure dataset. This indicates that the quality of the IMU data influences the accuracy of the foot velocity estimation.

For gait phase estimation (F1 score), our method showed strong performance overall, particularly excelling with synthetic IMU data from the AIST Gait Database. This outcome suggests synthetic IMU data more distinctly represents features such as ground contact states.

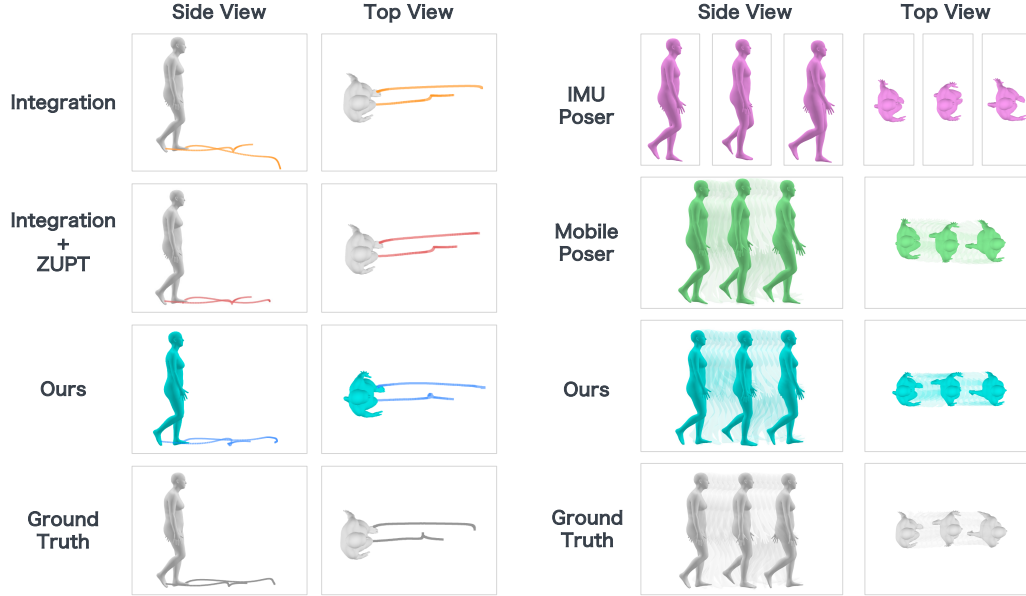


Figure 5.5: Qualitative comparison of foot trajectory (left) and full-body pose estimation (right) results using the AIST Gait DB. The results show side and top views for baseline methods (Integration, Integration + ZUPT, IMUPoser, MobilePoser) and Ours, compared with the Ground Truth.

**Qualitative Evaluation.** Figures 5.5, 5.6 and 5.7 illustrate the qualitative evaluation results. Figures 5.5 and 5.6 shows that the Integration and Integration+ZUPT methods resulted in unrealistic trajectories due to integration drift, such as floating above the ground or unnatural deviations. While ZUPT correction addressed unnatural vertical deviations and floating, it was insufficient in mitigating lateral drift relative to the walking direction. In contrast, our proposed method generated trajectories remarkably close to the ground truth, confirming stable trajectory estimation. This demonstrates that our method effectively employs anatomical constraints via human body modeling, significantly suppressing drift.

In Figure 5.7, full-body pose estimations from IMUPoser (without root translation), MobilePoser (with root translation), and our method (with root translation) were compared. All methods performed reasonably well



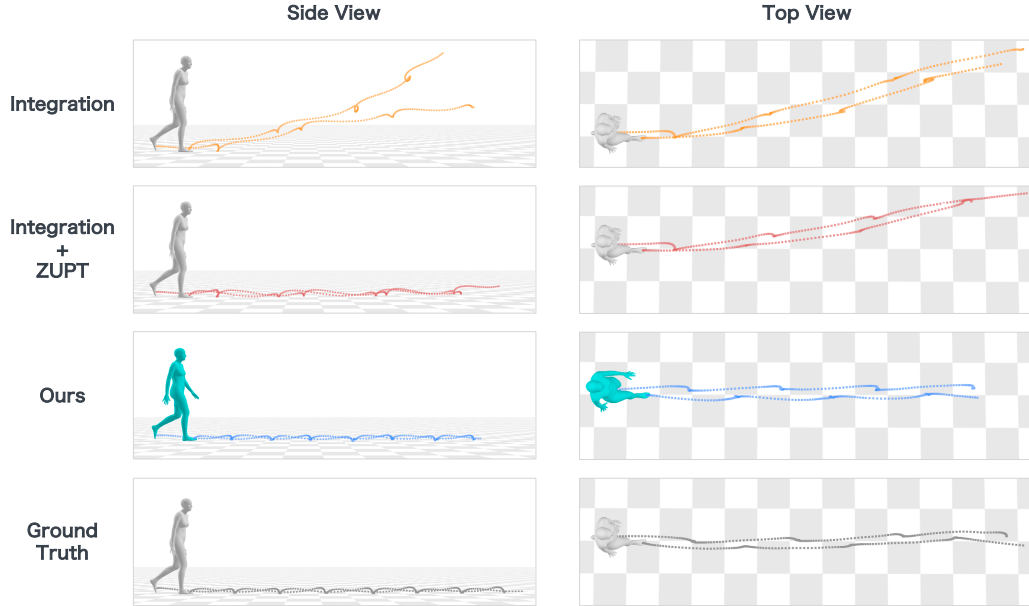


Figure 5.6: Qualitative comparison of foot trajectory estimation results on the UnderPressure. The side and top views illustrate estimated foot trajectories for Integration, Integration + ZUPT, Ours, and the Ground Truth.

in pose estimation; however, MobilePoser displayed noticeable directional errors, causing lateral drift in trajectories. Conversely, our method accurately reproduced the actual walking direction and trajectories, demonstrating superior directional estimation performance. This advantage arises because our method uses IMU data directly in the sensor coordinate system, thereby avoiding cumulative errors typically encountered with global coordinate transformations, particularly evident during extended measurements such as those captured by the UnderPressure dataset.

Overall, the quantitative and qualitative evaluations validate that our proposed method, utilizing only two IMUs placed on the foot soles, can achieve highly accurate gait motion estimation compared to existing methods. Clear advantages include effective drift suppression and stable estimation of walking direction, speed, and pose.

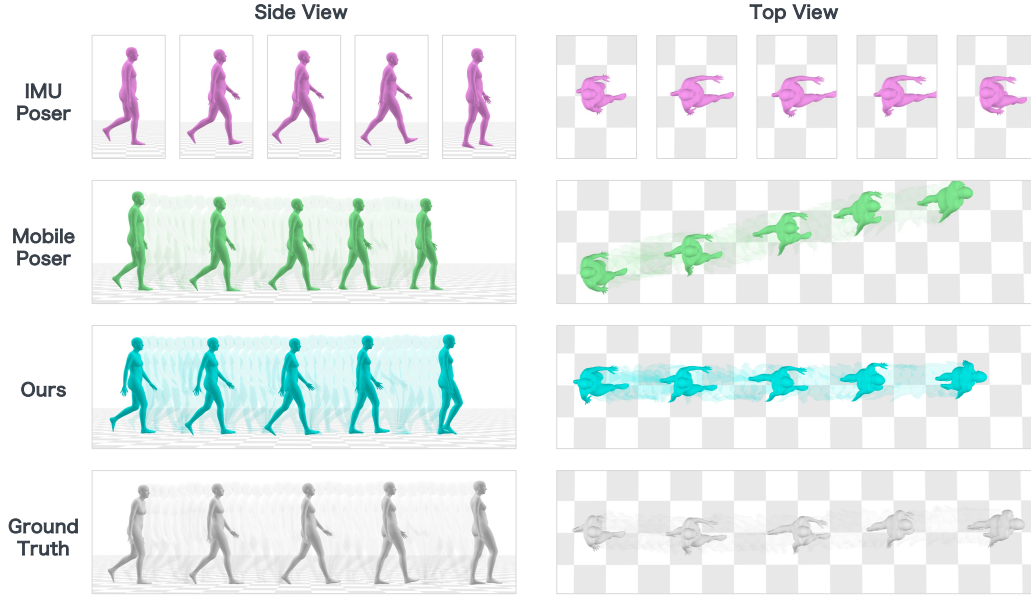


Figure 5.7: Qualitative comparison of full-body pose estimation results on the UnderPressure. Side and top views show results from IMUPoser, MobilePoser, Ours, and the Ground Truth. Compared to baseline methods, our approach achieves more accurate and natural full-body pose reconstructions, demonstrating stable performance even with real IMU data.

**Relationship between Joint Position Error and IMU Placement.** Figure 5.8 illustrates the distribution of joint position errors for each method in the UnderPressure dataset. Observing the heatmaps at the top, we can clearly identify a general trend: the joint position error is smaller in areas where IMUs are attached (indicated by pink stars), and increases as the distance from IMU attachment sites grows. This trend is consistent across all methods (IMUPoser, MobilePoser, and Ours), demonstrating that the direct motion information obtained from IMU-attached areas improves joint position estimation accuracy.

Examining the box plots at the bottom of Figure 5.8 more specifically, IMUPoser and MobilePoser exhibit relatively higher accuracy for the left arm where an IMU is attached, while accuracy tends to degrade for the

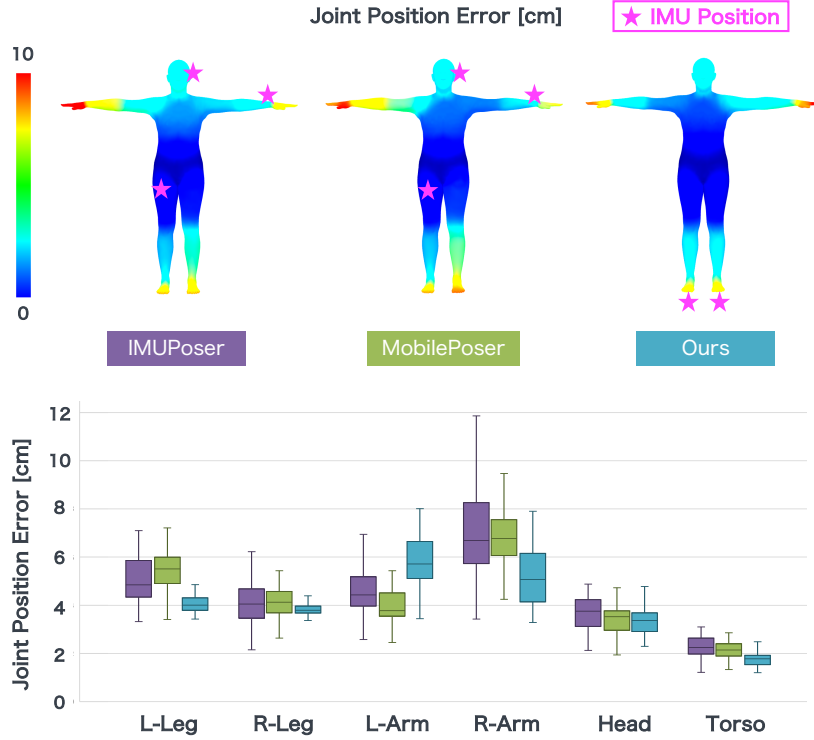


Figure 5.8: Visualization of joint position error (PEL-MPJPE) distributions. **Top:** Heatmaps illustrating that joints closer to IMU attachment sites (indicated by pink stars) exhibit lower positional errors. **Bottom:** Box plots detailing joint position errors for different body segments. Our method, utilizing IMUs attached to the soles of both feet, achieves notably higher accuracy in the lower limbs, essential for gait analysis.

right arm and lower limbs. In contrast, our proposed method, which attaches IMUs to the soles of both feet, significantly reduces joint position errors in the lower limbs, crucial for gait analysis. This confirms that attaching IMUs to both feet effectively captures lower limb motion more directly and accurately. As for the torso and head regions, there was little difference in accuracy among methods, suggesting that for straight-line walking movements, these regions are less affected by IMU placement.

These results clearly highlight the influence of IMU sensor placement

on the estimation accuracy of limb joint positions. Particularly for gait analyses, attaching IMUs to the foot soles proves highly effective, indicating that the sensor configuration adopted in our proposed method provides superior performance for lower-body motion estimation. On the other hand, the present comparison involves not only differences in IMU placement but also differences in the pose estimation mechanisms employed by each method. Therefore, the observed accuracy differences across methods cannot be attributed solely to variations in IMU configuration. To properly evaluate the contribution of the proposed estimation mechanism itself, it is necessary to conduct comparative experiments with existing methods under identical IMU placement conditions.

### 5.3.7 Comparison under Identical IMU Placement

In this section, we conduct a comparative experiment under conditions where all methods use two IMUs attached to the soles of the feet, enabling a fair comparison under identical sensor placement. The quantitative results are summarized in Table 5.2. The proposed method consistently outperforms baseline approaches on both the AIST Gait Database and the UnderPressure dataset across all metrics. In particular, the proposed method achieves strong performance in pose- and shape-related metrics, such as MPJPE, PEL-MPJPE, PVE-P, PVE-T, and JointAngle, demonstrating accurate estimation of both body pose and geometry.

The relatively lower performance of MobilePoser on the UnderPressure dataset, especially in the MPJPE and FootTraj metrics, can be attributed to the use of real insole-embedded IMUs with 6DoF sensing, which lack magnetometers and therefore cannot provide absolute yaw orientation. Although VQF is employed to transform IMU measurements into a global coordinate system, residual yaw drift and noise remain, leading to accumulated positional errors. In contrast, the proposed method directly uti-

Table 5.2: Comparison under identical IMU placement (two IMUs on foot soles). Bold indicates the best-performing method for each metric.

| Method                | MPJPE<br>[cm] (↓) | PEL-MPJPE<br>[cm] (↓) | PVE-P<br>[cm] (↓) | PVE-T<br>[cm] (↓) | JointAngle<br>[degree] (↓) | InterFoot<br>[cm] (↓) | FootTraj<br>[cm] (↓) | FootVel<br>[cm/s] (↓) | RootVel<br>[cm/s] (↑) | Phase<br>[F1] (↑) |
|-----------------------|-------------------|-----------------------|-------------------|-------------------|----------------------------|-----------------------|----------------------|-----------------------|-----------------------|-------------------|
| AIST Gait Database    |                   |                       |                   |                   |                            |                       |                      |                       |                       |                   |
| IMUPoser (2IMUs)      | -                 | 3.40                  | 4.45              | 4.40              | 2.83                       | 4.84                  | -                    | -                     | -                     | -                 |
| MobilePoser (2IMUs)   | 16.58             | 3.37                  | 4.39              | 4.40              | 2.93                       | 4.37                  | 17.46                | 55.84                 | 26.10                 | -                 |
| Ours (2IMUs)          | <b>5.73</b>       | <b>3.18</b>           | <b>4.36</b>       | <b>1.58</b>       | <b>2.26</b>                | <b>3.11</b>           | <b>5.32</b>          | <b>22.84</b>          | <b>8.25</b>           | <b>0.96</b>       |
| UnderPressure Dataset |                   |                       |                   |                   |                            |                       |                      |                       |                       |                   |
| IMUPoser (2IMUs)      | -                 | 7.36                  | 9.22              | 4.37              | 3.79                       | 5.92                  | -                    | -                     | -                     | -                 |
| MobilePoser (2IMUs)   | 47.36             | 6.67                  | 8.18              | 4.37              | 3.63                       | 6.39                  | 49.60                | 42.97                 | 34.02                 | -                 |
| Ours (2IMUs)          | <b>10.51</b>      | <b>3.87</b>           | <b>4.84</b>       | <b>2.06</b>       | <b>2.29</b>                | <b>4.25</b>           | <b>12.35</b>         | <b>26.70</b>          | 8.45                  | 0.89              |

lizes IMU measurements in the sensor coordinate system, thereby avoiding error accumulation caused by global coordinate transformations.

### 5.3.8 Ablation Study

To analyze the contribution of each component of the proposed architecture, we conduct an ablation study by comparing three configurations: the full model (Ours B+G+S), a variant without the Body Module (Ours G+S), and a variant without the Smoothing Module (Ours B+G). The quantitative results of this ablation study are summarized in Table 5.3.

The configuration without the Body Module (Ours G+S) exhibits degraded performance in metrics related to body shape and pose estimation, such as MPJPE, PVE, and InterFoot. This result indicates that estimating and incorporating a personalized body model from user attributes is important for accurately determining body-part positions and stride length. Removing the Smoothing Module (Ours B+G) leads to a noticeable degradation in velocity-related metrics, particularly FootVel and RootVel. This demonstrates that the Smoothing Module plays a key role in stabilizing temporal motion estimates.

Table 5.3: Ablation study of the proposed method, where B, G, and S denote the Body, Gait, and Smoothing Modules, respectively.

| Configuration                | MPJPE<br>[cm] (↓) | PEL-MPJPE<br>[cm] (↓) | PVE-P<br>[cm] (↓) | PVE-T<br>[cm] (↓) | JointAngle<br>[degree] (↓) | InterFoot<br>[cm] (↓) | FootTraj<br>[cm] (↓) | FootVel<br>[cm/s] (↓) | RootVel<br>[cm/s] (↑) | Phase<br>[F1] (↑) |
|------------------------------|-------------------|-----------------------|-------------------|-------------------|----------------------------|-----------------------|----------------------|-----------------------|-----------------------|-------------------|
| <b>AIST Gait Database</b>    |                   |                       |                   |                   |                            |                       |                      |                       |                       |                   |
| Ours G+S                     | 6.93              | 4.13                  | 5.17              | 3.10              | 2.23                       | 3.70                  | 8.48                 | 24.45                 | 8.92                  | 0.96              |
| Ours B+G                     | <b>4.79</b>       | <b>3.18</b>           | <b>4.35</b>       | <b>1.58</b>       | <b>2.22</b>                | 3.32                  | <b>4.28</b>          | 50.18                 | 8.84                  | 0.96              |
| Ours B+G+S                   | 5.73              | <b>3.18</b>           | 4.36              | <b>1.58</b>       | 2.26                       | <b>3.11</b>           | 5.32                 | <b>22.84</b>          | <b>8.25</b>           | 0.96              |
| <b>UnderPressure Dataset</b> |                   |                       |                   |                   |                            |                       |                      |                       |                       |                   |
| Ours G+S                     | 11.38             | 4.60                  | 5.41              | 2.97              | 2.38                       | 6.89                  | 15.24                | 29.33                 | <b>8.18</b>           | 0.90              |
| Ours B+G                     | 11.21             | 4.27                  | 5.37              | <b>2.06</b>       | 2.48                       | 4.74                  | 13.28                | 93.12                 | 10.58                 | 0.89              |
| Ours B+G+S                   | <b>10.51</b>      | <b>3.87</b>           | <b>4.84</b>       | <b>2.06</b>       | <b>2.29</b>                | <b>4.25</b>           | <b>12.35</b>         | <b>26.70</b>          | 8.45                  | 0.89              |

The effect of the Smoothing Module depends on the characteristics of the dataset. Since the AIST Gait Database is a synthetic dataset with minimal noise, the Gait Module alone already achieves high estimation accuracy, and applying smoothing can slightly degrade performance in some metrics. In contrast, for real-world data such as the UnderPressure dataset, which contains sensor noise and inter-window estimation errors, combining the Gait and Smoothing Modules improves both stability and accuracy. These results indicate that the proposed architecture is designed to function effectively on real-world data.

Figure 5.9 illustrates the Foot Velocity Error of the estimated poses obtained from the Gait Module and the Smoothing Module over two consecutive windows. In this experiment, the Gait Module estimates gait motion in 50-frame segments from the UnderPressure dataset, while the Smoothing Module processes overlapping segments shifted by half the window length (25 frames). The Gait Module alone exhibits discontinuities around window boundaries, which appear as large peaks in velocity error near the center of the sequence. By contrast, the Smoothing Module effectively eliminates these discontinuities and, together with the loss design described in Section 5.2.3, reduces velocity error across the entire sequence.

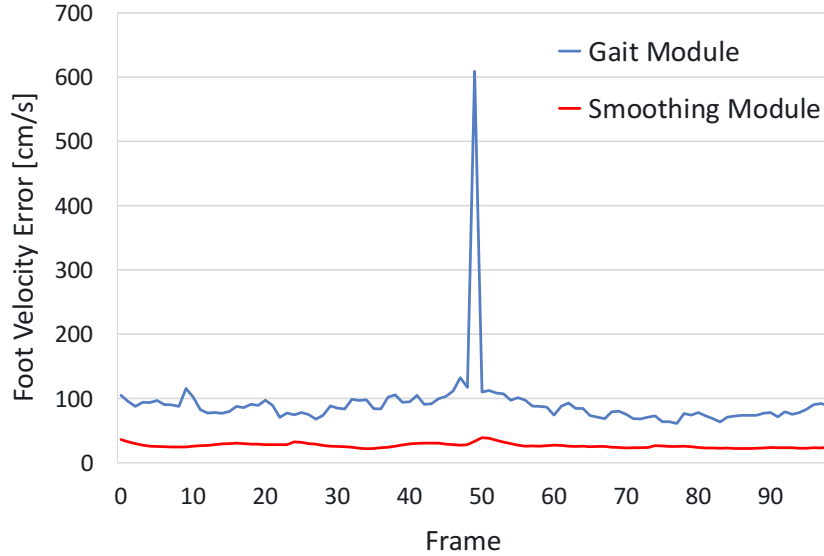


Figure 5.9: Foot Velocity Error of estimated poses over two consecutive windows, comparing the Gait Module (blue) and Smoothing Module (red). The results are computed on the UnderPressure dataset.

## 5.4 Discussion and Future Work

### 5.4.1 Practical Demonstration System

To demonstrate the practical applicability of our proposed method, we constructed a demonstration system for gait motion estimation using consumer-grade IMU devices (Figure 5.10). Specifically, we adopted the ORPHE CORE [136], a commercial device consisting of specialized shoes with IMUs embedded in the soles. In our demonstration system, the IMU data captured by the ORPHE CORE sensors are transmitted via Bluetooth Low Energy (BLE) to a laptop PC, allowing instantaneous estimation of gait poses. However, BLE communication is highly susceptible to interference from environmental factors, and transmitting data directly from the IMU sensors to a remote PC can be challenging due to limited communication

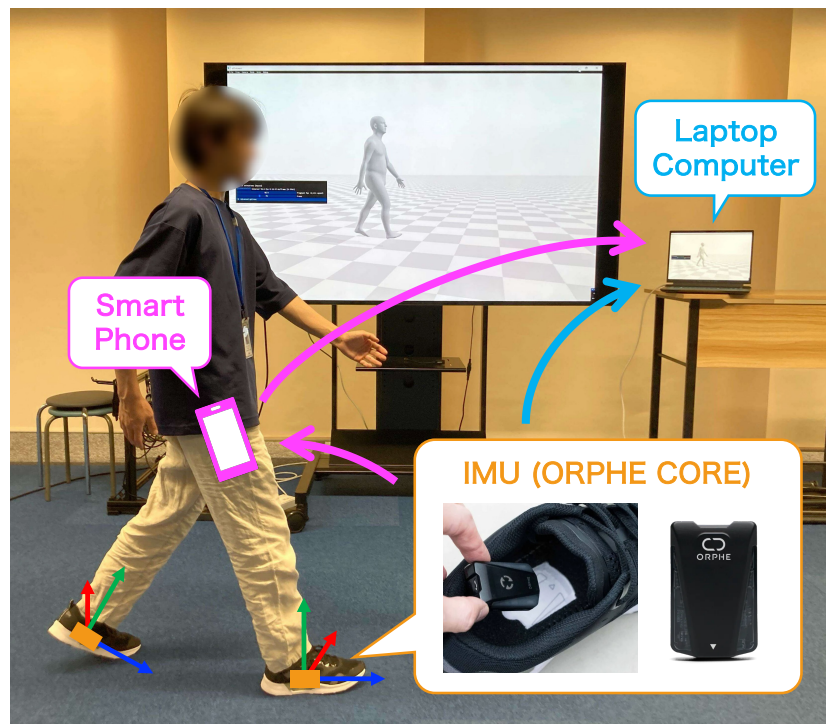


Figure 5.10: Demonstration system of our proposed method using consumer-grade IMU devices (ORPHE CORE). IMU data captured by sensors embedded in shoe soles are transmitted either directly to a laptop computer via Bluetooth (blue arrow), or relayed through a smartphone (magenta arrows) to minimize BLE interference. This setup ensures reliable gait pose estimation in practical scenarios.

range. To address this, assuming users typically carry their smartphones in hand or in a pocket, we developed an Android application to relay IMU data. As the smartphone remains in close proximity to the ORPHE CORE sensors, it significantly reduces data loss due to BLE interference. The collected data is then transmitted from the smartphone to the PC via Wi-Fi, enabling reliable operation even when the PC is not near the user.

Beyond ensuring robust data transmission, it is equally important for real-world applications that the estimation can be performed under limited



Table 5.4: Quantitative comparison of estimation accuracy across various motions, including straight walking at multiple speeds (Normal, Slow, Fast), non-straight walking, and jogging in the UnderPressure dataset.

| Method            | MPJPE<br>[cm] ( $\downarrow$ ) | PEL-MPJPE<br>[cm] ( $\downarrow$ ) | JointAngle<br>[degree] ( $\downarrow$ ) | RootVel<br>[cm/s] ( $\downarrow$ ) |
|-------------------|--------------------------------|------------------------------------|-----------------------------------------|------------------------------------|
| Normal Walk       | 9.20                           | 3.68                               | 2.24                                    | 8.16                               |
| Slow Walk         | 7.66                           | 3.87                               | 2.47                                    | 7.67                               |
| Fast Walk         | 13.86                          | 3.80                               | 2.28                                    | 9.27                               |
| Non-Straight Walk | 50.18                          | 11.97                              | 4.24                                    | 36.09                              |
| Jogging           | 73.87                          | 9.66                               | 10.24                                   | 48.48                              |

computational resources. The proposed model contains 3.05 M parameters and requires about 1.94 M FLOPs per frame, achieving an inference speed of 255 FPS on a workstation running Ubuntu 22.04.5 with an AMD EPYC 7453 (28 cores) CPU and an NVIDIA RTX A6000 GPU. By contrast, the baseline method MobilePoser has 6.67 M parameters, requires about 6.70 M FLOPs per frame, and achieves 93.2 FPS. These results demonstrate that the proposed method is lighter and faster than the baseline. An important future direction is to further reduce the model size and improve efficiency so that real-time operation can be achieved even on smartphones and other mobile or low-power devices.

### 5.4.2 Handling Straight-Line Walking Constraints

Our proposed method accurately estimates gait pose from minimal IMU data with instantaneous computation. However, the current model is trained specifically on straight-line walking. Consequently, when a user performs motions other than straight-line walking, unnatural movements may be estimated. Table 5.4 presents the estimation accuracy for three straight-walking speeds (Normal, Slow, and Fast) as well as for non-straight walk-

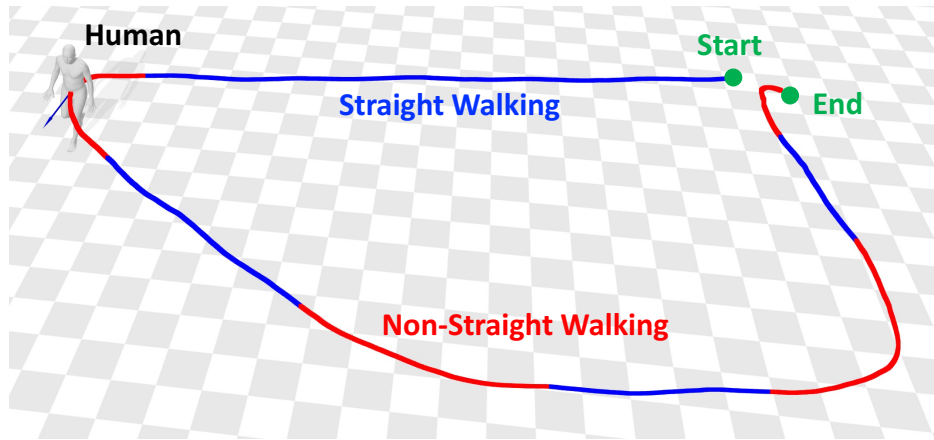


Figure 5.11: Example of straight-walking segmentation in the UnderPressure dataset. Blue indicates straight-walking segments and red indicates non-straight segments. Only the straight-walking segments were used for training and evaluation of the proposed method.

ing and jogging. For the three straight-walking speeds, although some speed-dependent variation is observed in metrics related to walking speed such as MPJPE and RootVel, the estimation maintains reasonable accuracy, confirming that the proposed method can adequately handle changes in walking speed. In contrast, for non-straight walking, examples of which are illustrated by the red trajectories in Figure 5.11, accuracy drops substantially, particularly in metrics other than joint angle estimation, and jogging shows decreased accuracy across all metrics. These results indicate that turning motions, such as those represented by the red line that exceed the range of body sway observed during straight walking (blue line) in Figure 5.11, as well as jogging motions, produce acceleration and angular velocity waveforms that fall outside the distribution of the training data, and thus the current model does not yet support such movements.

Moreover, during the training phase, the straight-line segments of the walking data must be manually extracted, and the demonstration system currently lacks a mechanism to identify straight-line walking segments in

real-time. To address this limitation, future development should focus on implementing real-time classification of IMU waveform data to automatically detect straight-line walking segments. By incorporating this functionality, we could naturally and continuously collect long-term gait data in everyday life without requiring conscious effort from the user, ultimately facilitating detailed analysis of long-term gait changes due to aging or medical conditions.

### 5.4.3 Addressing Device and Footwear Variability

Practical deployment of our method requires robustness to real-world factors that can affect sensing and estimation accuracy. Variations in footwear type and IMU attachment position may introduce measurement inconsistencies even for identical gait patterns. Possible alternative IMU placements, such as on the instep (e.g., shoelaces), shoe sides, or the heel, would lead to different sensor orientations and measurements, potentially degrading model performance due to inconsistencies in the input–output relationship. Moreover, attaching IMUs to soft shoe parts can introduce additional sensor noise from foot contact-induced vibrations. To mitigate these effects, we intentionally adopted the ORPHE CORE device embedded inside the shoe sole, ensuring a completely fixed and vibration-resistant position that minimizes sensitivity to placement variability.

Similarly, footwear type can influence IMU signals through differences in material stiffness, geometry, and fit, causing waveform variations even for the same motion. To reduce such variability and ensure consistent evaluation, we standardized footwear to sneakers, which are common in everyday use and specifically designed to house the ORPHE CORE device for robust and reproducible IMU placement. While this standardized setup enables highly accurate motion estimation, real-world usage may deviate from these conditions. Future work should therefore explore adaptive cali-

bration strategies and model generalization techniques to maintain reliable performance across diverse footwear and sensor placement conditions.

#### 5.4.4 Toward Real-World Gait Analysis

Our proposed method has demonstrated that gait motion can be accurately estimated from only two foot-mounted IMUs under controlled conditions of straight-line walking on level ground. However, comprehensive gait analysis in real-world conditions requires validating the proposed method in naturalistic settings, such as outdoor walkways, residential spaces, and other daily-life environments, to ensure reliable operation beyond controlled laboratory conditions. Long-term, in-the-wild evaluations will be critical to assess performance across ground surfaces of varying compliance (e.g., asphalt, concrete, grass, sand), thereby bridging the gap between laboratory demonstrations and everyday application.

In addition, comprehensive gait analysis in real-world conditions necessitates capturing and analyzing gait in diverse environments, including stairs and slopes. Developing robust algorithms capable of accurately estimating gait pose in these non-flat environments is an important future challenge. Leveraging multiple wearable sensors tailored to the specific movements and environmental contexts under investigation could further enhance motion estimation accuracy and versatility. Promising future directions include integrating insole foot-pressure sensors to perform biomechanical-informed pose estimation, employing smartwatches for detailed full-body motion reconstruction, and using first-person wearable cameras for combined environmental reconstruction and analysis of human interactions with surroundings. By integrating these complementary technologies, we anticipate the development of more comprehensive, versatile gait analysis methods that can be effectively applied across various aspects of everyday life.

### 5.4.5 Toward Clinical Applications

In this study, we proposed a method aimed at accurately reproducing normal gait motions of healthy individuals, but for future clinical applications two aspects will be crucial: constructing detailed, personalized human body models that represent individual patients, and accurately estimating pathological or otherwise abnormal gait patterns.

In this work we employed the SMPL model, a statistical body model built from the body shapes of healthy individuals, which assumes normal body morphology and therefore struggles to represent patients whose body structures have changed. For example, patients with knee osteoarthritis (KOA) are known to exhibit unique skeletal and gait characteristics caused by joint degeneration, pain, and muscular atrophy, features that standard statistical models may fail to capture. To address such clinical scenarios, it will be important to reconstruct high-resolution 3D personalized human body models from medical imaging data such as CT or MRI scans, allowing more accurate reproduction of pathological body structures and motions. Incorporating such personalized models will enable quantitative analysis of disease-specific gait patterns and is expected to contribute to clinical applications such as diagnosis, longitudinal monitoring, and treatment evaluation.

Furthermore, acquiring the ability to estimate abnormal gait unique to non-healthy populations is equally indispensable. Examples include asymmetric gait caused by hemiparesis, shuffling gait associated with Parkinson's disease, and spastic gait observed in cerebral palsy. To capture these conditions, it will be necessary to build and train on datasets that reflect the characteristic motion patterns of each disease. Since these pathological gaits often differ markedly from those of healthy individuals in foot-pressure distribution and muscle activation patterns, combining complementary sensing modalities such as insole pressure sensors or electromy-

graphy is also expected to be effective. Advancing such data-driven model extensions and sensor integration will enable high-accuracy joint motion estimation and quantitative characterization of disease-specific gait, further supporting diagnostic assistance and the evaluation of rehabilitation outcomes in clinical practice.

In addition, collecting large, well-annotated datasets of disease-specific gait is inherently challenging in clinical settings. To mitigate this data scarcity, it will be important to draw on strategies surveyed by Alzubaidi et al. [2], including domain-specific transfer learning, self-supervised representation learning, deep generative approaches such as GANs and DeepSMOTE, and physics-informed neural networks (PINNs). Building on these insights to design new learning frameworks tailored to wearable sensor data, such as IMU and insole pressure signals, will be crucial for robustly modeling patient-specific body structures and pathological gait patterns and for translating our method to future clinical applications.

## 5.5 Summary

In this chapter, we introduced “Gait Inertial Poser (GIP)”, a novel method for accurately estimating full-body human poses using only two shoe-embedded IMUs. Our gait-aware deep learning framework effectively addresses common IMU-based issues such as integration drift and orientation inaccuracies, without relying on global coordinate transformations. Experimental results on the AIST Gait Database and UnderPressure datasets demonstrated that our approach consistently outperforms baseline methods across multiple metrics. Additionally, an ablation study confirmed the significant contribution of our Smoothing Module in enhancing velocity estimation stability. We also developed a demonstration system utilizing commercially available IMU shoes (ORPHE CORE), verifying stable and near-instantaneous gait estimation in real-world settings. These findings high-

light the practical potential of IMU-based motion capture for long-term monitoring in everyday environments.

On the other hand, motion capture systems intended for purposes beyond long-term gait analysis, such as sports and biomechanical motion analysis, VR and AR applications, or robotics, must be able to handle a wide variety of full-body movements and interactions with environments that are not limited to flat terrain. However, IMUs measure only sparse inertial signals such as accelerations and angular velocities, which makes it difficult to estimate absolute positions and inevitably leads to drift. In addition, when only a small number of IMUs are available, it becomes fundamentally difficult to infer full-body motion because the sensors provide limited information. To address these challenges while keeping the sensing burden minimal, the next chapter proposes a method that reconstructs physically consistent full-body motion by integrating only four wearable devices, namely smartwatch IMUs and insole pressure sensors, within a physics simulation framework.

## Chapter 6

# Physics-based Pose Estimation

In this chapter, we propose Ground Reaction Inertial Poser (GRIP), a method that reconstructs physically plausible human motion using four wearable devices. Unlike conventional IMU-only approaches, GRIP combines IMU signals with foot pressure data to capture both body dynamics and ground interactions. Furthermore, rather than relying solely on kinematic estimation, GRIP uses a digital twin of a person, in the form of a synthetic humanoid (an articulated skeletal rigid-body model with human-like kinematic and dynamic properties) in a physics simulator, to reconstruct realistic and physically plausible motion. At its core, GRIP consists of two modules: KinematicsNet, which estimates body poses and velocities from sensor data, and DynamicsNet, which controls the humanoid in the simulator using the residual between the KinematicsNet prediction and the simulated humanoid state. To enable robust training and fair evaluation, we introduce a large-scale dataset, Pressure and Inertial Sensing for Human Motion and Interaction (PRISM), that captures diverse human motions with synchronized IMUs and insole pressure sensors. Experimental results show that GRIP outperforms existing IMU-only and IMU–pressure fusion methods across all evaluated datasets, achieving higher global pose accuracy and improved physical consistency.



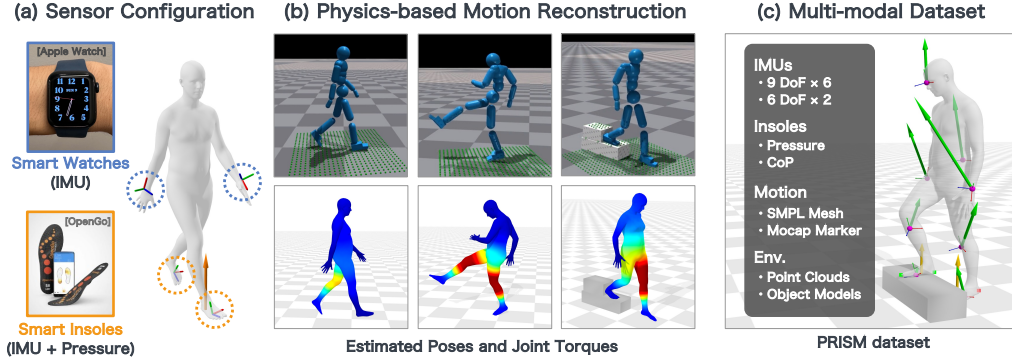


Figure 6.1: Overview of the proposed Ground Reaction Inertial Poser (GRIP). (a) GRIP observes motion using four IMUs and foot pressure data from smartwatches and smart insoles. (b) Full-body motion is reconstructed by driving a humanoid with joint torques in a physics simulator. (c) The PRISM dataset offers multimodal measurements, including IMUs, foot pressure, motion data, and environmental data.

## 6.1 Background

Accurately capturing human motion in practical settings is crucial for many applications in robotics, biomechanics, and virtual reality. Reducing the number of sensors is particularly important for improving usability in everyday environments. In this work, we propose **Ground Reaction Inertial Poser (GRIP)**, a novel motion capture method that reconstructs physically consistent full-body motion using only four wearable devices attached to the wrists and feet (Fig. 6.1-a). GRIP integrates kinematic cues from four IMUs with dynamic cues from foot pressure, and leverages physics simulation-based humanoid control through deep reinforcement learning, achieving accurate and physically plausible full-body motion estimation despite the minimal sensor configuration (Fig. 6.1-b).

Conventional optical motion capture (MoCap) systems provide high accuracy, but require dedicated environments with multiple cameras, making them unsuitable for everyday or long-term use. Monocular RGB-based

approaches have also been widely studied in recent years. However, third-person monocular methods [11,20,27,129,141] are constrained by the camera's field of view, while egocentric methods [35,44,48,54,140,152] suffer from the difficulty of estimating the camera motion and the full-body pose due to self-occlusions. As an alternative to these vision-based approaches, wearable IMU-based methods, in which sensors are directly attached to the human body, have also been explored. Commercial wearable MoCap systems offer accurate motion estimation, but systems such as Xsens require up to 17 IMUs embedded in a spandex body suit, making them less suitable for daily use.

Recent studies address this limitation by estimating full-body pose using only a few IMUs [46,59,95,156]. However, unlike methods that leverage VR-oriented head-mounted devices (HMDs) or hand trackers for pose estimation [1,14,22,25,26,29,55–57,71,142,172], IMU-only systems cannot directly measure absolute position. As a result, accumulated estimation errors lead to temporal drift and physically implausible artifacts, such as foot sliding, penetration, or floating. Although post-hoc physical optimization techniques [149,153,155,157] can mitigate these issues, they still depend on IMU-based estimation of body–environment contacts, making it difficult to accurately capture fine-grained physical interactions. To better model such interactions, recent studies have incorporated insole-type pressure sensors equipped with IMUs [145,159], providing additional dynamic cues that reflect weight shifts through ground reaction forces, as well as reliable foot-contact cues that help suppress drift. Nevertheless, these approaches rely solely on foot sensors, limiting their ability to reconstruct upper-body motion and still suffer from drift in complex or dynamic motions.

To address these challenges, this work designs GRIP to achieve accurate and physically plausible full-body motion estimation using a minimal configuration of IMUs and foot pressure sensors. Here, the term “mini-

mal” refers to a sensor placement that is both necessary to make generic human motion uniquely identifiable in principle and feasible for practical deployment. Since the motion of terminal body parts, such as the hands and feet, is fundamentally difficult to estimate without corresponding sensors except for periodic motions, this work adopts a configuration with four IMUs attached to the wrists and ankles, together with foot pressure sensors on both feet. This sensor configuration enables the integrated estimation of lower-body physical interactions, such as foot contact and load transfer, along with full-body motion including the upper limbs. Moreover, this configuration is primarily intended for motion monitoring in realistic environments, such as industrial and caregiving or rehabilitation settings, where sensors are worn as part of routine tasks or measurements, and is therefore considered practically feasible.

GRIP consists of two neural network modules. The first, KinematicsNet, incrementally estimates the kinematic state of motion, including joint positions, angles, and velocities, from IMU and foot pressure data. The second, DynamicsNet, controls a digital twin humanoid model within a physics simulator by using the sensor data, the estimated kinematic state, humanoid state, and environmental information as observations to reproduce physically consistent human motion. By using the simulator, DynamicsNet naturally satisfies physical constraints such as gravity, friction, and ground reaction forces. This allows it to suppress non-physical artifacts, as well as unnatural foot-ground constraints seen in the physical optimization methods.

Validating such a physics-integrated framework requires real-world data that capture both human motion and its physical interactions with the environment. To this end, we built a new public dataset, Pressure and Inertial Sensing for Human Motion and Interaction (**PRISM**), which integrates multi-modal data including IMU and insole pressure measurements, optical motion capture, and environmental models (Fig. 6.1-c). PRISM covers

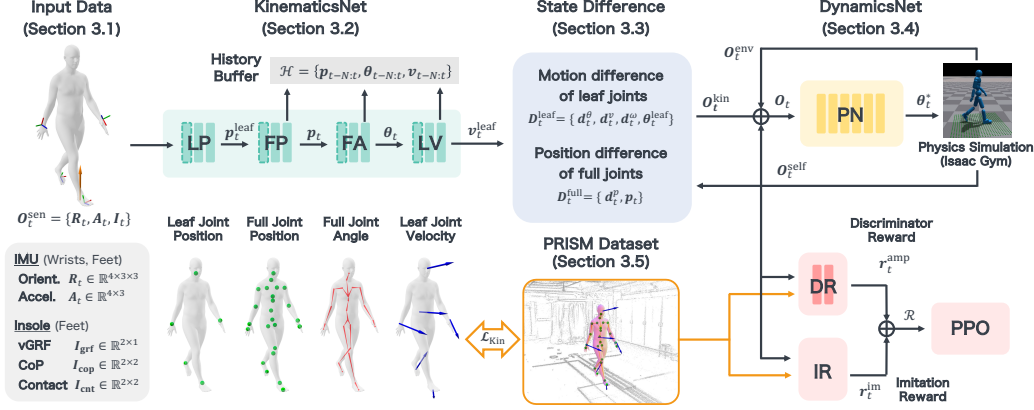


Figure 6.2: Overview of the GRIP framework. **Input Data** (Sec. 6.2.1) consists of IMU and insole measurements. **KinematicsNet** (Sec. 6.2.2) estimates kinematic states, and the **State Difference** (Sec. 6.2.3) compares them with the simulated humanoid. **DynamicsNet** (Sec. 6.2.4) drives the humanoid through physics simulation-based control. The **PRISM dataset** (Sec. 6.2.5) provides diverse multi-modal data.

diverse human motions, spanning daily activities, sports, and human–object interactions, offering a comprehensive dataset for modeling physically consistent human motion and interaction.

In summary, our contributions are as follows: (1) We propose GRIP, a framework that achieves accurate full-body motion estimation using only four IMUs and insole pressure sensors. (2) We introduce a two-stage architecture, KinematicsNet and DynamicsNet, that integrates IMU and pressure data to reconstruct physically consistent human motion. (3) We construct and publicly release a new multi-modal dataset, PRISM, which includes diverse actions and sensor modalities.

## 6.2 Method

Our goal is to reproduce diverse real-world human motions on a torque-driven humanoid model in a physics simulator using a minimal sensor setup. As shown in Fig. 6.2, GRIP takes four IMU signals, two on the wrists and two embedded in the insoles, along with the foot pressure data. The architecture follows an observer–controller structure: KinematicsNet estimates motion states from sensor data, and DynamicsNet uses these estimates to control the humanoid model. We introduce an intermediate representation, State Difference, which captures the discrepancy between the estimated motion state and the simulated humanoid state, allowing DynamicsNet to utilize the kinematic observations. The following sections describe the input data (Sec. 6.2.1), KinematicsNet (Sec. 6.2.2), State Difference (Sec. 6.2.3), DynamicsNet (Sec. 6.2.4), and the PRISM dataset (Sec. 6.2.5).

### 6.2.1 Input Data

The GRIP framework uses acceleration and orientation measurements from four IMUs. All IMU accelerations and orientations are transformed into the global coordinate frame. For a sequence of length  $T$ , the acceleration and orientation at frame  $t$  are represented as  $\mathbf{A}_t \in \mathbb{R}^{4 \times 3}$  and  $\mathbf{R}_t \in \mathbb{R}^{4 \times 3 \times 3}$ . The insole sensors contain 16 pressure cells per foot and provide raw pressure values, vertical ground reaction forces (GRF), and the center of pressure (CoP). To ensure compatibility across different insole devices, we define a simplified input representation. Specifically, the vertical GRF is represented as  $\mathbf{I}_{\text{grf}} \in \mathbb{R}^{2 \times 1}$ , the CoP positions (2D coordinates centered at the midfoot) as  $\mathbf{I}_{\text{cop}} \in \mathbb{R}^{2 \times 2}$ , and the binary contact labels for the forefoot and rearfoot regions as  $\mathbf{I}_{\text{cnt}} \in \mathbb{R}^{2 \times 2}$ . The contact labels are obtained by dividing the foot sole into front and rear regions and assigning a contact label when the summed pressure in each region exceeds a threshold.

These features are concatenated to form the simplified input representation  $I_t = [I_{\text{grf}}, I_{\text{cop}}, I_{\text{cnt}}] \in \mathbb{R}^{2 \times 5}$ . Finally, all sensor observations at frame  $t$  are summarized as  $O_t^{\text{sen}} = \{R_t, A_t, I_t\}$ .

### 6.2.2 KinematicsNet

KinematicsNet progressively estimates kinematic information from the input sensor data. The network produces four outputs: leaf-joint positions (LP), full-joint positions (FP), full-body joint angles (FA), and leaf-joint velocities (LV). First, taking the root joint (pelvis) as the coordinate origin, the network predicts the leaf-joint positions (LP) of the wrists, feet, and head. These predictions, together with the sensor data, are then used to reconstruct full-body joint positions (FP). While this progressive estimation follows prior works [153, 155–157], our method differs in that we operate directly on globally oriented IMU measurements. Because no IMU is attached to the root joint, the root’s global rotation cannot be separated from these measurements. As a result, the network outputs are represented in a root-centered frame but still retain the global rotation of the body. Next, using the sensor data and the estimated joint positions, the network predicts the full-body joint angles (FA) and the global leaf-joint velocities (LV). The resulting full-joint positions, joint angles, and leaf-joint velocities are used to compute the State Difference (Sec. 6.2.3), and are also stored in a History Buffer  $\mathcal{H}$  to provide reset references when the humanoid falls in the physics simulator (Sec. 6.2.4).

The network employs a unidirectional LSTM for each stage (LP, FP, FA, LV), estimating outputs frame by frame while modeling temporal dependencies. At each timestep, the output is defined as leaf-joint position  $p_t^{\text{leaf}} \in \mathbb{R}^{5 \times 3}$ , full-joint position  $p_t \in \mathbb{R}^{24 \times 3}$ , full-body joint angle  $\theta_t \in \mathbb{R}^{24 \times 6}$ , and leaf-joint velocity  $v_t^{\text{leaf}} \in \mathbb{R}^{5 \times 3}$ . The 24 joints correspond to the SMPL model [79], and the five leaf joints correspond to the wrists, feet, and head.

The joint angles are represented using the continuous 6D rotation representation [174] for stable learning. KinematicsNet is trained using mean squared error (MSE) losses applied to each of its submodules. The total loss is defined as:  $\mathcal{L}_{\text{Kin}} = \|\mathbf{p}^{\text{leaf}} - \hat{\mathbf{p}}^{\text{leaf}}\|^2 + \|\mathbf{p} - \hat{\mathbf{p}}\|^2 + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2 + \|\mathbf{v}^{\text{leaf}} - \hat{\mathbf{v}}^{\text{leaf}}\|^2$ , where the hat notation indicates ground-truth values.

### 6.2.3 State Difference

In IMU-based motion estimation, integrating estimated velocities to obtain global trajectories causes drift accumulation, which would directly propagate to humanoid tracking. To avoid this, our framework does not use integrated global positions; instead, it represents translational motion using the estimated global velocities of leaf joints and expresses posture using root-relative joint positions. To unify these cues and capture the discrepancy between the estimated and simulated humanoid states, we introduce an intermediate representation called State Difference.

The State Difference comprises (1) motion differences at the leaf joints and (2) position differences for all joints, both transformed into the humanoid's heading-aligned coordinate frame. The first component,  $\mathbf{D}_t^{\text{leaf}} = \{\mathbf{d}_t^\theta, \mathbf{d}_t^v, \mathbf{d}_t^\omega, \boldsymbol{\theta}_t^{\text{leaf}}\}$ , consists of the following terms: the rotational differences  $\mathbf{d}_t^\theta \in \mathbb{R}^{4 \times 6}$ , angular velocity differences  $\mathbf{d}_t^\omega \in \mathbb{R}^{4 \times 3}$ , and orientations  $\boldsymbol{\theta}_t^{\text{leaf}} \in \mathbb{R}^{4 \times 6}$  computed at the four leaf joints corresponding to the IMUs (wrists and feet); and the linear velocity differences  $\mathbf{d}_t^v \in \mathbb{R}^{6 \times 3}$  computed at six key joints (wrists, feet, head, and pelvis) based on the outputs of KinematicsNet. The second component,  $\mathbf{D}_t^{\text{full}} = \{\mathbf{d}_t^p, \mathbf{p}_t\}$ , captures the position differences of all joints, expressed in the root-aligned coordinate frame. Specifically, the position difference  $\mathbf{d}_t^p \in \mathbb{R}^{24 \times 3}$  are derived from the full-body joint positions  $\mathbf{p}_t \in \mathbb{R}^{24 \times 3}$  estimated by KinematicsNet and the corresponding simulated humanoid joint positions. Finally,  $\mathbf{D}_t^{\text{leaf}}$  and  $\mathbf{D}_t^{\text{full}}$  are concatenated to form the State Difference vector  $\mathbf{D}_t = [\mathbf{D}_t^{\text{leaf}}, \mathbf{D}_t^{\text{full}}]$ .

### 6.2.4 DynamicsNet

DynamicsNet controls the humanoid model within the physics simulator based on several observations. The overall framework is formulated as a Markov Decision Process defined by the tuple  $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ , where  $\mathcal{S}$ ,  $\mathcal{A}$ ,  $\mathcal{T}$ ,  $\mathcal{R}$ , and  $\gamma$  denote the state space, action space, transition dynamics, reward function, and discount factor, respectively. At each timestep  $t$ , the policy  $\pi(\mathbf{a}_t | \mathbf{s}_t)$  selects an action  $\mathbf{a}_t \in \mathcal{A}$  given the current state  $\mathbf{s}_t \in \mathcal{S}$ , and the simulator updates the next state  $\mathbf{s}_{t+1}$  according to  $\mathcal{T}$ . The policy is trained to maximize the expected discounted reward using the proximal policy optimization (PPO) algorithm [116].

**Observations.** The observation vector at each timestep is defined as  $\mathbf{O}_t = \{\mathbf{O}_t^{\text{sen}}, \mathbf{O}_t^{\text{kin}}, \mathbf{O}_t^{\text{self}}, \mathbf{O}_t^{\text{env}}\}$ . The sensor observation  $\mathbf{O}_t^{\text{sen}} = \{\mathbf{R}_t, \mathbf{A}_t, \mathbf{I}_t\}$  consists of the same inputs used in KinematicsNet. These sensor-level features allow the policy to directly reference the original sensory inputs in addition to the estimated kinematic states. The kinematic observation  $\mathbf{O}_t^{\text{kin}}$  corresponds to the State Difference  $\mathbf{D}_t$  introduced in the previous section, representing the discrepancies between the estimated motion and simulated humanoid states. The self observation  $\mathbf{O}_t^{\text{self}} = \{\mathbf{p}_t^{\text{sim}}, \boldsymbol{\theta}_t^{\text{sim}}, \dot{\mathbf{p}}_t^{\text{sim}}, \dot{\boldsymbol{\theta}}_t^{\text{sim}}\}$  includes the positions and angles of all joints, along with their temporal derivatives (velocities and angular velocities). Finally, the environmental observation  $\mathbf{O}_t^{\text{env}}$  represents the height map of the environment, sampled as a  $25 \times 25$  grid centered around the humanoid within a square area of  $1.5\text{m} \times 1.5\text{m}$ . This map provides spatial context, enabling the policy to adapt to non-flat environments.

**Policy and Reward.** The policy network (PN) is implemented as a multi-layer perceptron (MLP) that maps the observation vector  $\mathbf{O}_t$  to target joint angles  $\boldsymbol{\theta}_t^*$ , which serve as the action  $\mathbf{a}_t$ . These target angles are converted into joint torques  $\boldsymbol{\tau}_t$  using proportional-derivative (PD) control:  $\boldsymbol{\tau}_t = k_p(\boldsymbol{\theta}_t^* -$



$\theta_t) - k_d \dot{\theta}_t$ , where  $k_p$  and  $k_d$  denote proportional and derivative gains, respectively. The reward function follows the Perpetual Humanoid Control (PHC) framework [86] and consists of three components: an Adversarial Motion Prior (AMP) [103], an imitation reward, and an energy penalty [30]. The total reward is formulated as:

$$r_t = 0.5r_t^{\text{amp}} + 0.5r_t^{\text{imit}} + r_t^{\text{energy}}, \quad (6.1)$$

where  $r_t$  denotes the total reward at timestep  $t$ .

The AMP reward  $r_t^{\text{amp}}$  encourages realistic and natural motions by training a discriminator  $D$  to distinguish generated motions from real human motion. The discriminator takes as input the self-observation  $O_t^{\text{self}}$ , which consists of a temporally ordered sequence of motion features over  $W$  consecutive frames, maintained as a sliding window that is updated at every timestep. The AMP reward is computed as:

$$r_t^{\text{amp}} = -\log \left( 1 - \sigma(D(O_t^{\text{self}})) \right), \quad (6.2)$$

where  $\sigma(\cdot)$  denotes the sigmoid function and  $D(O_t^{\text{self}})$  represents the discriminator logit output for the self-observation  $O_t^{\text{self}}$ . This reward formulation encourages the policy to generate motions that the discriminator classifies as real, i.e.,  $\sigma(D(O_t^{\text{self}})) \rightarrow 1$ , thereby maximizing  $r_t^{\text{amp}}$ . The discriminator is trained using a standard Generative Adversarial Network (GAN) objective with gradient penalty regularization:

$$\begin{aligned} \mathcal{L}_D = & \mathbb{E}_{O_{\pi}^{\text{self}}} \left[ \log \left( 1 - \sigma(D(O_{\pi}^{\text{self}})) \right) \right] \\ & + \mathbb{E}_{O_{\text{ref}}^{\text{self}}} \left[ \log \sigma(D(O_{\text{ref}}^{\text{self}})) \right] \\ & + \lambda_{\text{gp}} \mathbb{E}_{O_{\text{ref}}^{\text{self}}} \left[ \left\| \nabla_O D(O_{\text{ref}}^{\text{self}}) \right\|^2 \right], \end{aligned} \quad (6.3)$$

where  $O_{\pi}^{\text{self}}$  denotes self-observations generated by the policy,  $O_{\text{ref}}^{\text{self}}$  denotes self-observations from the reference motion dataset,  $\lambda_{\text{gp}}$  is the gradient

penalty coefficient, and  $\nabla_O D(\cdot)$  represents the gradient of the discriminator output with respect to the input observation.

The imitation reward measures the consistency between the simulated and reference motion states. It is computed based on the differences in joint positions, rotations, linear velocities, and angular velocities:

$$\begin{aligned} r_t^{\text{imit}} = & w_p e^{-100\|\hat{\mathbf{p}}_t - \mathbf{p}_t\|} + w_\theta e^{-100\|\hat{\boldsymbol{\theta}}_t \ominus \boldsymbol{\theta}_t\|} \\ & + w_v e^{-10\|\hat{\mathbf{v}}_t - \mathbf{v}_t\|} + w_\omega e^{-0.1\|\hat{\boldsymbol{\omega}}_t - \boldsymbol{\omega}_t\|}, \end{aligned} \quad (6.4)$$

where  $w_p$ ,  $w_\theta$ ,  $w_v$ , and  $w_\omega$  are weighting coefficients for each term, and  $\ominus$  denotes the rotation difference between the reference and simulated orientations. The hatted quantities (e.g.,  $\hat{\mathbf{p}}_t$ ,  $\hat{\boldsymbol{\theta}}_t$ ) represent reference motion values, while the unaccented ones correspond to simulated states.

The energy penalty  $r_t^{\text{energy}}$  is computed as the negative mechanical power of the humanoid, obtained from the element-wise product of joint torques  $\boldsymbol{\tau}_t$  and joint angular velocities  $\boldsymbol{\omega}_t$ . Specifically, the penalty is defined as

$$r_t^{\text{energy}} = -\alpha |\boldsymbol{\tau}_t \odot \boldsymbol{\omega}_t|, \quad (6.5)$$

where  $\alpha$  is a power coefficient (default 0.0005). This term encourages energy-efficient control while excluding the first three frames to avoid initialization artifacts.

**Fall Recovery Mechanism.** In our framework, the humanoid is controlled under strong physical constraints without introducing any auxiliary forces such as residual forces. As a result, during complex or abrupt motion transitions, the humanoid may occasionally lose balance and fall, leading to unstable simulation behavior. To address this issue, we introduce a fall recovery mechanism based on a history buffer to ensure stable inference during deployment.

The mechanism maintains a history buffer  $\mathcal{H} = \{\mathbf{p}_{t-N:t}, \boldsymbol{\theta}_{t-N:t}, \mathbf{v}_{t-N:t}\}$  containing the past  $N$  frames of outputs from KinematicsNet. At each

timestep  $t$ , a fall is detected when the root joint height  $\mathbf{p}_{z,t}^{\text{root}}$  falls below the threshold  $\tau_z$  and the discriminator probability  $\rho_t$  from the AMP module is lower than  $\tau_\rho$ . Once a fall is detected, we compute the kinematic root displacement  $\Delta \mathbf{p}_{t-N:t}^{\text{kin, root}}$  by integrating the root velocities estimated by KinematicsNet and stored in the buffer. The simulated root position is then reinitialized as  $\mathbf{p}_t^{\text{sim, root}} = \mathbf{p}_{t-N}^{\text{sim, root}} + \Delta \mathbf{p}_{t-N:t}^{\text{kin, root}}$ . At the same time, the joint angles  $\boldsymbol{\theta}_t$  estimated by the Full-joint Angle (FA) module of KinematicsNet are used to reset the humanoid states, after which the physical simulation is resumed. In addition, during the falling phase, the corresponding segment of motion output is replaced by the buffered KinematicsNet predictions  $\mathbf{p}_{t-N:t}$ ,  $\boldsymbol{\theta}_{t-N:t}$ , and  $\mathbf{v}_{t-N:t}$  to maintain kinematic continuity and prevent corrupted physical transitions. This allows the humanoid to continue its motion smoothly and consistently even after a fall, without requiring external correction forces.

### 6.2.5 PRISM Dataset

To enable robust training and fair evaluation of GRIP, we require data captured under diverse conditions with multiple IMUs, insole pressure sensors, and high-fidelity motion labels. However, existing pressure-based human motion datasets [39, 96, 117, 159, 164] are limited in sensor modalities, motion diversity, or pose annotation accuracy, making them insufficient for training and evaluating physically consistent pose estimation from sparse sensors. To address these limitations, we constructed a new multimodal dataset, Pressure and Inertial Sensing for Human Motion and Interaction (PRISM), that covers a broad set of motions with synchronized multi-modal sensor data. This section describes the details on sensor specifications, calibration, synchronization, and comparisons with existing datasets.

**Data Acquisition.** In constructing the PRISM dataset, it is necessary to integrate the various sensor measurements obtained from multiple wear-

able devices and an optical MoCap system into a unified global coordinate frame and to synchronize all recordings under a consistent temporal reference. This section describes the sensing devices and data acquisition protocol used during collection, the coordinate-frame calibration procedures for each device, and the temporal synchronization process. The study was formally approved by the Institutional Review Board (IRB) at Carnegie Mellon University under approval number STUDY2022\_00000178.

**Sensors and Capture Protocol.** Data were collected using an optical motion-capture system (OptiTrack), IMU-equipped smartwatches (Apple Watch), IMU-equipped insole-type pressure sensors (Moticon OpenGo), a head-mounted multimodal sensing device (Meta Aria Glass), and IMU devices (Xsens MTw). OptiTrack provides high-precision 3D marker trajectories, and the captured marker data were fitted with Mosh++ [80] to reconstruct SMPL meshes and high-fidelity body motion. OpenGo includes a 6DoF IMU (accelerometer and gyroscope) and sixteen pressure sensors on each foot, enabling the measurement of vertical forces, pressure distribution, and center of pressure (CoP). The Apple Watch includes a 9DoF IMU (accelerometer, gyroscope, and magnetometer). Aria Glass includes two 9DoF IMUs and simultaneously records forward-facing video and two additional SLAM camera views. Using these videos and IMU data, the Project Aria Machine Perception Service (MPS) was used to obtain environmental point clouds and device trajectories. Furthermore, Xsens MTw units (9DoF IMUs) were attached to the knees and pelvis.

At the beginning of each recording, the subject maintained a T-pose for approximately three seconds while wearing all sensors. This static posture is used to determine the relative orientations of each IMU with respect to the SMPL body model. Starting each capture with the same protocol also reduces ambiguity in the initial pose during real-world operation. After the T-pose, the subject returned to an upright stance and performed three

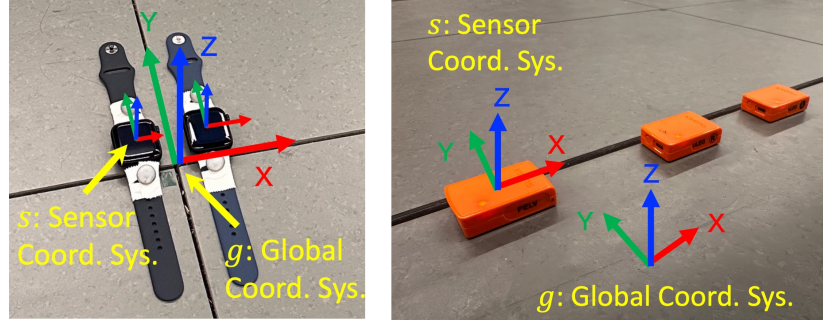


Figure 6.3: Calibration setup for initial orientation. Left: Apple Watch. Right: Xsens MTw sensors.

consecutive vertical jumps. The characteristic acceleration peaks observed during the jump motions are later used as robust temporal landmarks for precise synchronization across all sensor streams. The subject then performed the instructed motion sequence.

**IMU Calibration.** IMU data were collected from eight body locations using four types of wearable devices: Apple Watch, OpenGo, Aria Glass, and Xsens MTw. Because each device defines its measurements in its own coordinate system, all IMU observations must be transformed into a unified global coordinate frame. We use the coordinate system of the SMPL motion reconstructed from the optical MoCap data using Mosh++ as the global coordinate frame, adopting a z-up convention. At the beginning of each recording session, the subject holds a T-pose facing the positive  $y$ -axis of the global frame. This protocol enables consistent initialization of all IMU orientations and establishes the reference posture used for the subsequent calibration steps. The following sections describe how raw IMU orientations and accelerations are aligned with the global coordinate frame and further mapped to the corresponding SMPL joint frames.

We first describe the calibration procedure for the Apple Watch and Xsens IMUs, following the notation used in previous works [59, 70]. The

relationship between the joint-frame orientation and the raw IMU measurements is expressed as follows:

$$\mathbf{R}_g^j = \mathbf{R}_g^r \mathbf{R}_r^s \mathbf{R}_s^j, \quad \mathbf{a}_g = \mathbf{R}_g^r \mathbf{R}_r^s \mathbf{a}_s, \quad (6.6)$$

where  $r$  denotes the raw IMU reference frame, and  $g$  denotes the global coordinate frame. To obtain the joint rotations and accelerations in the global coordinate system at each IMU attachment site, the IMU orientation must be converted into the corresponding joint orientation  $\mathbf{R}_g^j$ , and the raw accelerations defined in the sensor frame must likewise be transformed into the global frame  $\mathbf{a}_g$ , where  $\mathbf{R}_r^s$  denotes the raw IMU orientation expressed in the reference sensor frame. Thus, we first estimate the orientation of this reference frame relative to the global frame. Following prior work [59,70], before being attached to the user, each IMU sensor is placed on the floor in a specified orientation aligned with the global frame and kept still for 3 seconds (Fig. 6.3). By averaging the static IMU signal, we obtain a stable estimate of its orientation at that moment, where  $\mathbf{R}_r^s = \mathbf{R}_r^g$  holds. The inverse  $(\mathbf{R}_r^g)^{-1}$  is then applied to compute  $\mathbf{R}_g^r$ . In the next step, we compute the joint orientation  $\mathbf{R}_g^j$  by combining the IMU-to-global rotation with the joint-to-sensor rotation as  $\mathbf{R}_g^s \mathbf{R}_s^j$ . To determine the joint-to-sensor transform  $\mathbf{R}_s^j$ , the subject takes a T-pose while facing the global  $y$ -axis. The relationship is given by

$$\mathbf{R}_{sT}^{jT} = (\mathbf{R}_r^{sT})^{-1} \mathbf{R}_r^g \mathbf{R}_g^{jT}, \quad (6.7)$$

where the subscript  $T$  indicates values obtained during the T-pose calibration. Assuming the IMU remains rigidly attached to the body during capture, the relative transform  $\mathbf{R}_s^j$  is constant, allowing us to treat  $\mathbf{R}_{sT}^{jT} = \mathbf{R}_s^j$ . Because the joint rotation in the T-pose,  $\mathbf{R}_g^{jT}$ , is determined from the SMPL model, the T-pose raw IMU measurements  $\mathbf{R}_r^{sT}$  provide a reliable estimate of  $\mathbf{R}_s^j$ .

Next, we describe the calibration procedure for the OpenGo device. Unlike the other IMU devices, the sensor embedded in OpenGo is a 6DoF

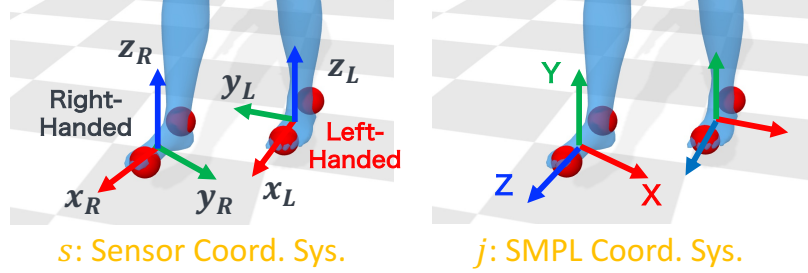


Figure 6.4: Sensor and global coordinate frames for the OpenGo devices, illustrating the handedness difference between sensors.

IMU without a magnetometer, and therefore cannot utilize magnetic-field information for orientation estimation. Following prior work [5], we use the Versatile Quaternion-based Filter (VQF) [69] to estimate the sensor orientation based on the gravity vector. Let the raw sensor coordinate frame be  $s$  and the SMPL foot-joint coordinate frame be  $j$ . Because the left and right OpenGo sensors follow different handedness conventions (Fig. 6.4), axis-sign corrections are applied to unify their coordinate systems. The measured angular velocity  $\omega_s$  and acceleration  $a_s$  are mapped into the joint frame using the fixed sensor-to-joint rotation  $R_j^s$  as

$$\begin{aligned}
 \omega_{j,\text{Right}} &= R_j^s \omega_s, \\
 \omega_{j,\text{Left}} &= R_j^s (\omega_s \odot [-1, 1, -1]), \\
 a_{j,\text{Right}} &= R_j^s a_s, \\
 a_{j,\text{Left}} &= R_j^s (a_s \odot [1, -1, 1]),
 \end{aligned} \tag{6.8}$$

where  $\odot$  denotes the element-wise product. VQF provides the foot orientation in its gravity-aligned frame, denoted as  $R_{\text{vqf}}^j$ . To express this orientation in the global coordinate frame, we align it with the SMPL foot-joint orientation in the T-pose. Let  $R_g^{j^T}$  be the global SMPL foot-joint rotation in the T-pose, and let  $R_{\text{vqf}}^{j^T}$  be the VQF-estimated sensor orientation at the same moment. The global joint-frame orientation is obtained by directly aligning the VQF-estimated sensor orientation with the calibrated joint ori-

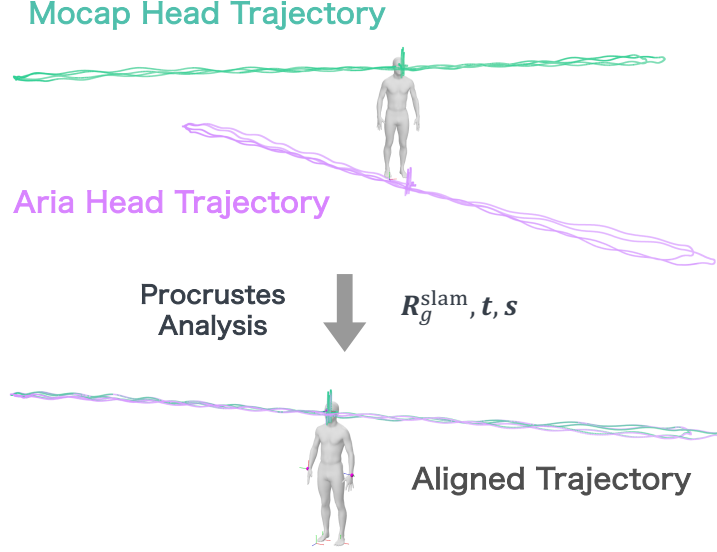


Figure 6.5: Alignment of the Aria Glass SLAM trajectory to the MoCap-based head trajectory using Procrustes analysis.

entation in the T-pose:

$$R_g^j = R_g^{jT} \left( R_{vqf}^{jT} \right)^{-1} R_{vqf}^j. \quad (6.9)$$

The global acceleration is obtained by rotating the raw sensor acceleration using the joint's global orientation:

$$a_g = R_g^j a_j. \quad (6.10)$$

Finally, we describe the calibration procedure for the Aria Glass. Aria Glass is equipped with two IMUs (1000 Hz and 800 Hz), and in our dataset we use the 1000 Hz unit, which is downsampled to 100 Hz to match the other sensors. The Aria system defines several coordinate frames, including the IMU frame  $i$ , the device frame  $d$ , the SLAM world frame  $w$  estimated by MPS, and the Central Pupil Frame (CPF)  $c$ , which is used to approximate the effective IMU attachment location. The global MoCap coordinate system is denoted as  $g$ . MPS provides the device orientation



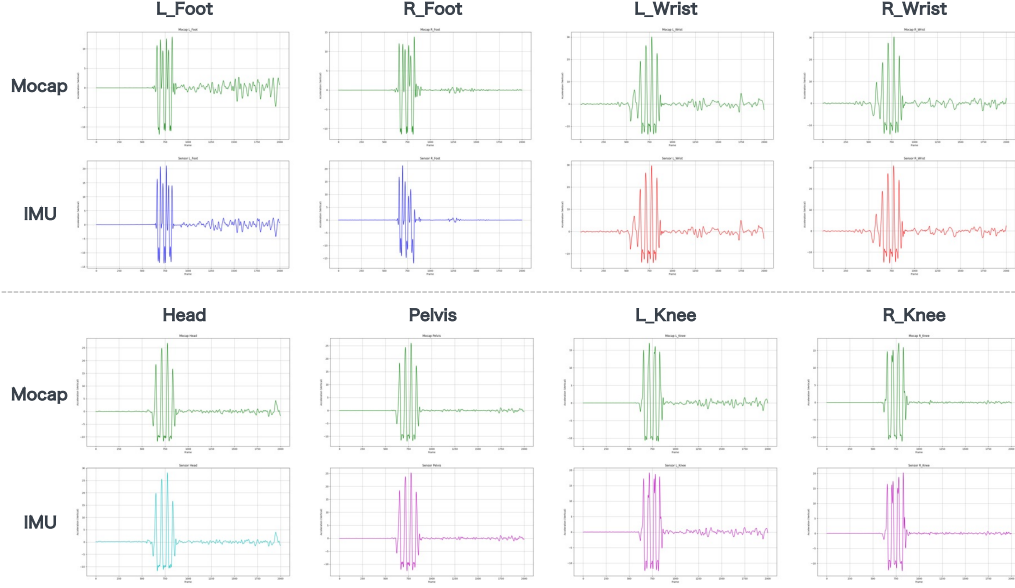


Figure 6.6: Vertical acceleration signals from MoCap data and from IMU sensors at the same set of body-mounted locations. The characteristic jump-induced peaks provide reliable cues for temporal synchronization across sensor streams.

as  $R_w^d$  in the SLAM coordinate system, and factory calibration further provides the fixed extrinsic rotations  $R_d^i$  (IMU to device) and  $R_i^c$  (IMU to CPF). The transformation from the SLAM frame to the global MoCap frame,  $R_g^w$ , is estimated by leveraging the high accuracy of the SLAM trajectory produced by MPS. Specifically, as illustrated in Fig. 6.5, we align the Aria Glass trajectory to the MoCap trajectory of the SMPL-based CPF (central pupil) position across the entire sequence using Procrustes analysis, which yields the rotation  $R_g^w$ , translation  $t$ , and scale  $s$ . Using these transformations, the CPF orientation and IMU accelerations can be expressed in the global coordinate system as

$$\begin{aligned} R_g^c &= R_g^w R_w^d R_d^i R_i^c, \\ a_g &= R_g^w R_w^d R_d^i a_i, \end{aligned} \quad (6.11)$$

where  $a_i$  denotes the raw acceleration measured in the IMU frame. These

quantities are used as the head IMU’s global orientation and global acceleration in the dataset.

**Time Synchronization.** All sensors used in this study (OptiTrack, Apple Watch, OpenGo, Aria Glass, and Xsens MTw) record data at 100 Hz. Since the measurements on each device are started manually, their recording start times are not aligned. Therefore, a synchronization procedure is required to align all sensor streams to a common temporal reference. In our dataset, we use the vertical acceleration peaks generated during the three vertical jumps performed immediately after the T-pose as temporal landmarks for synchronization. As described in the previous section, all IMUs are transformed into a unified global coordinate system through calibration. This allows the vertical acceleration components derived from each IMU’s measurements to be compared directly. Meanwhile, the SMPL mesh obtained from MoCap provides only positional information; therefore, the corresponding reference acceleration is computed by applying finite differences to the time-series vertex positions at each IMU attachment site, converting positions to velocities and subsequently to accelerations.

For synchronization, the vertical acceleration waveform of each IMU is slid over the reference SMPL-derived vertical acceleration waveform using a sliding-window approach, and similarity (cross-correlation) is evaluated at each offset. The temporal offset that achieves the highest similarity is taken as the time shift for that IMU stream, aligning all sensors to a common time axis. Fig. 6.6 shows an example of the vertical acceleration signals computed from MoCap-based SMPL at the eight IMU attachment locations (wrists, feet, knees, waist, and head), alongside those from the corresponding IMU sensors, illustrating how the characteristic peaks from the three jumps serve as reliable synchronization cues.

**Dataset Composition.** The PRISM dataset consists of 1,275 sequences di-

Table 6.1: Comparison of human motion datasets utilizing insole-type foot pressure sensors.

| Dataset             | Year | Foot IMUs | Other IMUs | Env. Models | Motion Diversity      | MoCap Method   |
|---------------------|------|-----------|------------|-------------|-----------------------|----------------|
| PSU-TMM100 [117]    | 2020 | No        | No         | No          | Low (Tai Chi)         | Optical        |
| UnderPressure [96]  | 2022 | Yes (2)   | No         | No          | Medium (Locomotion)   | Inertial       |
| MMVP [164]          | 2024 | No        | No         | No          | Medium (Exercise)     | RGBD           |
| HAMPI [159]         | 2025 | Yes (4)   | No         | No          | High (Diverse Action) | Inertial + RGB |
| <b>PRISM (Ours)</b> | 2025 | Yes (2)   | Yes (6)    | Yes         | High (Diverse Action) | Optical        |

vided into 1,021 for training and 254 for testing, each 10 seconds long (1,000 frames). Data were collected from six subjects (four male and two female), totaling approximately 12,750 seconds ( $\approx 3.5$  hours) of motion. Each sequence includes synchronized multimodal signals, such as MoCap-based reference poses, IMU and insole pressure data, environmental point clouds, camera trajectories, and object models for interaction.

The dataset covers a wide range of motion categories. First, Basic Locomotion includes walking along predefined paths and random trajectories (Walking Square / Walking Random), as well as jogging along predefined or random paths (Jogging Square / Jogging Random). Next, Slow Movements consist of stretching, two-foot and one-foot balancing, squatting, and lunge walking. Furthermore, Fast Sports Actions include whole-body athletic movements such as sidestepping, forward jumping, golf, baseball, tennis, soccer, and direction-switching movements. Finally, Interaction Behaviors involve movements that include physical interaction with objects, such as stepping onto boxes or stairs (Stepping Boxes / Stepping Stair), sitting on boxes (Sitting Boxes), and object carrying.

**Comparison with Existing Datasets.** In recent years, several foot-sensing motion datasets leveraging insole-type pressure sensors have been released, contributing significantly to the understanding of human motion. However, existing datasets exhibit inherent trade-offs across aspects such as

IMU availability, environmental information, motion diversity, and annotation accuracy. As a result, none of them is sufficient for comprehensively modeling physically consistent full-body motion together with interactions involving the environment and objects.

PSU-TMM100 [117] is a dataset that combines insole-type pressure sensors with optical motion capture, consisting of Tai Chi movements. While it provides plantar pressure distributions, it does not include any IMU measurements. UnderPressure [96] contains long recordings of basic locomotion such as walking, jogging, and skipping, and is characterized by the inclusion of both plantar pressure sensors and foot IMUs. It also includes a limited set of environment-interaction behaviors, such as stair ascent/descent and stepping onto objects, but neither environment geometry nor object models are provided. MMVP [164] records RGB-D video of body movements together with plantar pressure measurements, and features several fast exercise motions, but does not provide IMU data. HAMPI [159], the most recent foot-sensing dataset, includes plantar pressure, foot IMUs, and SMPL motions generated by combining inertial motion capture with monocular video-based pose estimation. Although it covers a wide variety of actions, the IMUs are restricted to the feet, making the dataset unsuitable for studies that require wrist or full-body IMUs. Moreover, some motion-capture errors remain in the form of unnatural body tilts or elbow twisting, making the annotations not fully reliable for physics-based approaches such as GRIP.

Taken together, existing datasets do not simultaneously provide (i) detailed foot-sensing measurements, (ii) full-body inertial information, (iii) environment interaction data, (iv) a broad range of motions, and (v) accurate full-body mesh annotations within a single framework. PRISM addresses these gaps by combining SMPL motion reconstructed from optical MoCap with eight IMUs placed at common attachment sites (knees, pelvis, wrists, and head) in addition to the feet, along with insole-type pressure

sensors. PRISM also includes environmental point clouds and object models, and covers a wide variety of motions ranging from basic locomotion to slow movements, fast sports activities, and interactions with objects. Through this combination, PRISM serves as a unified dataset suitable for integrated analysis of full-body motion and foot-ground interaction dynamics in everyday scenarios, supporting research that requires physically consistent motion estimation.

## 6.3 Experiments

### 6.3.1 Implementation Details

**Networks and Training.** Our framework consists of two stages: KinematicsNet, which estimates kinematic states, and DynamicsNet, which generates physically consistent motion within a physics simulator. Training is performed in two stages: first, KinematicsNet is trained independently, and then DynamicsNet is trained within a physics simulation environment. During the training of DynamicsNet, the parameters of KinematicsNet are frozen and used as a motion-state estimator.

KinematicsNet comprises four submodules: LP (leaf-joint position), FP (full-joint position), LV (leaf-joint velocity), and FA (full-joint angle). Each submodule is first trained independently to ensure stable estimation, and then all modules are jointly trained in an integrated manner. Following previous works [153, 155, 157, 168], we initialize the hidden states of each submodule RNN using the ground-truth data of the first frame, to avoid ambiguity caused by uncertain initial poses. This initialization is applied only to the first frame of each sequence. Training was conducted for 2000 epochs using the Adam optimizer (learning rate  $1 \times 10^{-5}$ , batch size 256). Following TIP [59], IMU signals were processed with a low-pass filter to remove high-frequency noise. All training was performed on data sampled

at 100 Hz.

DynamicsNet is a policy network trained within a physics simulation environment to control a torque-driven humanoid model. The policy network is implemented as a multilayer perceptron (MLP) consisting of six layers with sizes [2048, 1536, 1024, 1024, 512, 512] and SiLU activations. The network outputs target joint angles, which are converted into joint torques through PD control. In the early training stage, an early termination condition is applied: an episode is terminated when any joint position error exceeds a threshold of  $\tau_e = 0.25$  m, which accelerates convergence and stabilizes the learning of basic motion patterns. We train DynamicsNet for 500 epochs across 10 iterations under this early-termination regime using 4096 parallel environments, a policy minibatch size of 32,768, and an AMP minibatch size of 8192. Afterward, the early termination constraint is removed to enable training over long motion sequences, and an additional 500 epochs across 20 iterations are performed under the full-length setting. In addition, to enable fall-recovery mechanism during inference and allow the simulation to continue even after the humanoid falls, we introduce a fall-detection condition in which the humanoid is considered fallen when the root joint height drops below 0.30 m and the AMP discriminator probability  $\rho_t$  falls below 0.7.

**Computational Environment.** Experiments were conducted on Ubuntu 20.04.6 LTS. The system is equipped with an AMD EPYC 7543 processor configuration providing 64 cores (128 threads), 251 GB RAM, and eight NVIDIA RTX A6000 GPUs with 48 GB VRAM each. All neural networks were implemented in PyTorch and optimized using the Adam optimizer. Physics simulations and reinforcement learning were performed using Isaac Gym in a CUDA 11.8-enabled environment with NVIDIA driver version 570.133.07.

Table 6.2: Quantitative comparison of our method with baseline methods across the three datasets. Lower values indicate better performance for all metrics. Bold numbers denote the best performance for each metric, and underlined numbers denote the second best.

| Dataset        | Method             | MPJPE<br>[mm] (↓) | PEL-MPJPE<br>[mm] (↓) | PA-MPJPE<br>[mm] (↓) | MPJRE<br>[deg] (↓) | Acc<br>[m/s <sup>2</sup> ] (↓) | FS<br>[m/s] (↓) | FP<br>[mm] (↓) | vGRF<br>[N] (↓) |
|----------------|--------------------|-------------------|-----------------------|----------------------|--------------------|--------------------------------|-----------------|----------------|-----------------|
| PRISM          | PIP                | 248.59            | 85.48                 | <u>33.35</u>         | 17.08              | 6.68                           | <u>0.20</u>     | 10.71          | <b>246.85</b>   |
|                | GlobalPose         | <u>198.30</u>     | <b>43.50</b>          | <b>31.29</b>         | <b>12.01</b>       | 7.72                           | 0.22            | <u>9.72</u>    | 299.22          |
|                | MobilePoser        | 267.45            | 72.76                 | 55.69                | 17.99              | <b>6.20</b>                    | <b>0.19</b>     | 9.97           | <u>248.54</u>   |
|                | FoRM               | 199.60            | 87.34                 | 63.75                | 20.23              | 9.67                           | 0.36            | 15.64          | -               |
|                | SolePoser          | -                 | -                     | 82.75                | -                  | 9.68                           | -               | -              | -               |
|                | <b>GRIP (Ours)</b> | <b>182.44</b>     | <u>63.85</u>          | 46.47                | <u>13.89</u>       | <u>7.30</u>                    | 0.21            | <b>5.77</b>    | 258.40          |
| Under Pressure | PIP                | 523.65            | <u>29.89</u>          | 21.08                | 8.35               | <b>12.59</b>                   | <u>0.32</u>     | 1.43           | <u>265.62</u>   |
|                | GlobalPose         | <u>301.12</u>     | <b>21.49</b>          | <b>17.41</b>         | <b>7.40</b>        | 16.65                          | 0.32            | 3.31           | 287.12          |
|                | MobilePoser        | 626.62            | 44.28                 | 33.74                | 11.73              | <u>12.62</u>                   | 0.35            | <u>1.27</u>    | <b>244.78</b>   |
|                | FoRM               | 553.19            | 59.29                 | 32.60                | 18.50              | 17.34                          | 0.57            | 21.52          | -               |
|                | SolePoser          | -                 | -                     | 34.52                | -                  | 24.27                          | -               | -              | -               |
|                | <b>GRIP (Ours)</b> | <b>218.09</b>     | 37.27                 | 27.16                | <u>7.64</u>        | 13.22                          | <b>0.31</b>     | <b>0.00</b>    | 278.27          |
| PSU-TMM100     | PIP                | 182.14            | 87.56                 | 61.62                | 21.12              | 1.62                           | <b>0.09</b>     | 5.84           | 367.87          |
|                | GlobalPose         | 175.96            | <b>63.05</b>          | <b>50.28</b>         | <u>18.71</u>       | 2.09                           | 0.14            | <u>2.95</u>    | <u>340.11</u>   |
|                | MobilePoser        | 210.66            | 112.35                | 85.46                | 28.10              | 2.37                           | <u>0.10</u>     | 5.28           | 358.19          |
|                | FoRM               | <u>126.60</u>     | 98.02                 | 82.45                | 25.19              | <u>1.60</u>                    | 0.13            | 4.51           | -               |
|                | SolePoser          | -                 | -                     | 97.11                | -                  | <b>1.36</b>                    | -               | -              | -               |
|                | <b>GRIP (Ours)</b> | <b>118.60</b>     | <u>70.32</u>          | <u>55.72</u>         | <b>16.72</b>       | 4.31                           | 0.11            | <b>0.73</b>    | <b>316.06</b>   |

**Datasets.** We use three datasets with distinct motion characteristics: **PRISM**, **UnderPressure** [96], and **PSU-TMM100** [117]. PRISM contains diverse real-world motions, including daily activities, sports actions, and human-object interactions, captured with real IMUs and insole pressure sensors. UnderPressure primarily consists of locomotion-centric actions such as walking, jogging, and skipping; although some sequences include interactions such as stair climbing or stepping onto objects, these were excluded in our experiments because object models are not provided. PSU-TMM100 focuses on Tai Chi movements, which involve slow, highly controlled motions with rich balance shifts. Since PRISM includes only six subjects, we randomly split all sequences into an 8:2 ratio for training and testing. For UnderPressure and PSU-TMM100, which each contain ten subjects, eight were used for training and two for testing. Although both datasets provide real insole measurements, they do not contain IMU recordings; therefore, following prior work [156], we synthesized IMU signals by computing finite differ-

Table 6.3: Comparison of input modalities, output representations, and physical modeling across baseline methods.

| Method             | Input Data |        | Output Data |      |         | Physics |      |
|--------------------|------------|--------|-------------|------|---------|---------|------|
|                    | IMUs       | Press. | Joints      | Mesh | Transl. | Optim.  | Sim. |
| PIP [155]          | 6          |        | ✓           | ✓    | ✓       | ✓       |      |
| GlobalPose [153]   | 6          |        | ✓           | ✓    | ✓       | ✓       |      |
| MobilePoser [149]  | 3          |        | ✓           | ✓    | ✓       | ✓       |      |
| FoRM [159]         | 2          | ✓      | ✓           | ✓    | ✓       |         |      |
| SolePoser [145]    | 2          | ✓      | ✓           |      |         |         |      |
| <b>GRIP (Ours)</b> | 4          | ✓      | ✓           | ✓    | ✓       |         | ✓    |

ences of the vertex orientations at IMU attachment locations on the SMPL mesh. All datasets were evaluated in 500-frame (5-second) segments.

**Metrics.** We evaluate both the proposed and baseline methods in terms of kinematic accuracy and physical plausibility. For kinematic accuracy, we compute Mean Per Joint Position Error (**MPJPE**); Pelvis-Aligned MPJPE (**PEL-MPJPE**), which removes global translation; and Procrustes-Aligned MPJPE (**PA-MPJPE**), which further removes global scale and rotation. We also evaluate joint rotation accuracy using Mean Per Joint Rotation Error (**MPJRE**), and temporal smoothness using Acceleration Error (**Acc**). For physical plausibility, we measure Foot Sliding (**FS**) during contact and Foot Penetration Depth (**FP**) below the ground or object surface. To assess dynamic consistency, we compute the Vertical Ground Reaction Force Error (**vGRF**), applied to baselines with physical optimization and to our method. Metric units are reported in Table 6.2.

### 6.3.2 Comparison

We compare our method with baseline methods that estimate full-body motion using IMUs and insole pressure sensors. Table 6.3 summarizes the input modalities, output representations, and whether each method incor-



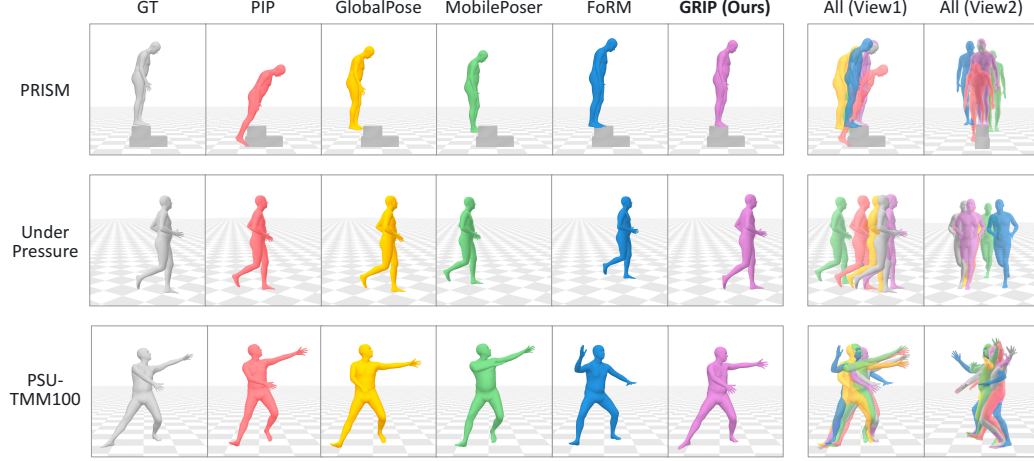


Figure 6.7: Qualitative comparison of pose estimation results across the three datasets. Our method accurately reconstructs foot placement on objects (PRISM), exhibits less position drift (UnderPressure), and captures slow weight-shifting motions (PSU-TMM100).

porates physical modeling. Quantitative results are presented in Table 6.2, and qualitative results are shown in Figs. 6.7, 6.8 and 6.9.

For global pose accuracy (MPJPE), our method achieves the best performance across all datasets, PRISM (diverse motions), UnderPressure (large displacements), and PSU-TMM100 (slow motions with weight shifts). The foot-sliding metric (FS) shows that physics-based approaches, including post-hoc optimization methods (PIP, GlobalPose, MobilePoser) and our simulation-based method, suppress foot sliding more effectively than the purely kinematic FoRM. However, optimization-based methods may over-apply contact constraints when estimation errors occur, temporarily locking the foot to the ground and resulting in trajectory errors (Fig. 6.8). In contrast, our method simulates natural foot-ground interactions, enabling physically plausible and accurate trajectories (Fig. 6.9). For pose metrics without global translation (PEL-MPJPE, PA-MPJPE, MPJRE), GlobalPose achieves the highest accuracy due to using six IMUs and a correction mechanism based on a pelvis-mounted IMU that learns gravity-pose relation-

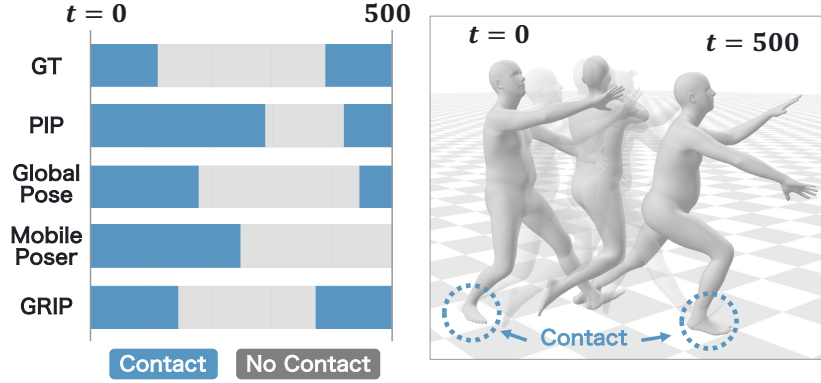


Figure 6.8: Comparison of foot contact timing. Right-foot contact labels during low-speed motions in PSU-TMM100, computed from the estimated GRF of the physics-based methods.

ships. PIP and our four-IMU method achieve comparable accuracy, as the use of pressure information effectively captures body dynamics and compensates for the absence of pelvis and head IMUs. SolePoser and FoRM, which rely only on insole-mounted IMUs and pressure signals, are inherently limited in reconstructing accurate full-body motion.

For the foot penetration metric (FP), our method achieves the lowest values across all datasets. In flat-ground datasets such as UnderPressure and PSU-TMM100, the three physics-optimization methods exhibit penetration only when contact detection fails, and our method also shows only minor penetration in limited situations, such as during state replacement during a fall. All four physics-based methods fundamentally prevent penetration through physical constraints. In contrast, the purely kinematic method FoRM is more prone to penetration caused by root-position estimation errors, particularly in highly dynamic sequences such as PRISM and UnderPressure. In the PRISM dataset, which contains objects, our method accurately handles foot-object interactions through a physics-based object model and achieves higher accuracy than MobilePoser and PIP, which assume flat-ground contact, and GlobalPose, which optimizes height and

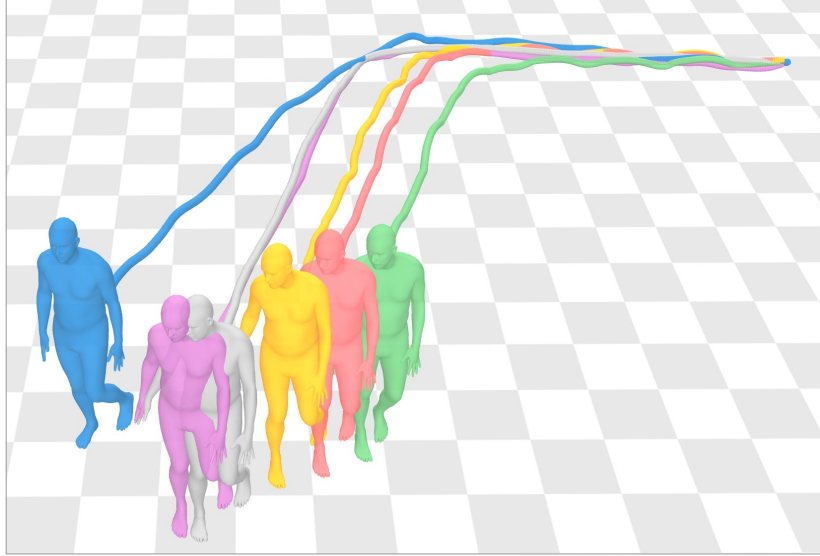


Figure 6.9: Qualitative comparison of estimated poses and root trajectories for a walking sequence from the UnderPressure dataset. Colors correspond to the same methods shown in Fig. 6.7.

support states but may still suffer from imperfect contact detection. However, in our method, errors may still occur in cases such as stumbling or stepping off elevated surfaces, where the estimated foot position temporarily falls below the object surface.

For vGRF error, PIP and MobilePoser achieve the highest accuracy in PRISM and UnderPressure, while our method performs best in PSU-TMM100. The three optimization-based methods adopt floating-base models and stabilize physics-based control by applying residual forces. In contrast, our method employs a humanoid model without floating bases or residual forces, which can lead to foot-pressure patterns that differ from those of real humans during balance loss or recovery in high-momentum motions. However, in slower motions such as those in PSU-TMM100, the other methods tend to miss fine-scale foot contact events and consequently produce larger ground-reaction and position errors. Our method maintains higher

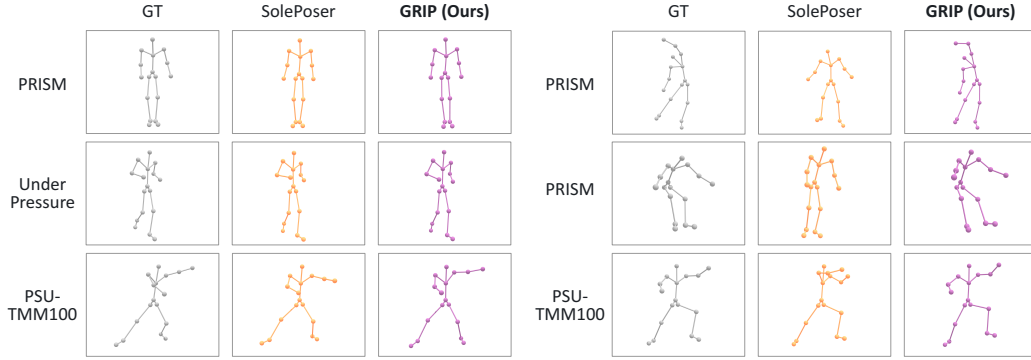


Figure 6.10: Comparison between SolePoser and our method using the same 17 joint locations extracted from the SMPL model.

accuracy in these cases by combining natural foot–ground interaction through physics simulation with pressure-driven dynamic reasoning. Further details and extended results can be found in the Supplementary Material.

We also present a qualitative comparison between SolePoser [145] and our method in Fig. 6.10. SolePoser produces reasonable estimates for relatively simple poses and periodic locomotion such as walking and jogging. However, due to the limitation of using only foot sensors, its accuracy degrades significantly for motions involving substantial upper-body movement or complex postural changes, and several failure cases can be observed. In contrast, GRIP incorporates wrist-mounted IMUs in addition to foot IMUs and plantar-pressure sensing, allowing it to capture a broader range of whole-body inertial information. As a result, our method reconstructs full-body poses more accurately even in non-periodic and complex motions where SolePoser struggles. These observations indicate that GRIP can estimate whole-body dynamics that cannot be recovered from foot-only sensing, by combining IMUs at the extremities (hands and feet) with foot pressure information.

### 6.3.3 Evaluation

**Sensor Configuration Analysis.** This experiment examines how different sensor configurations affect estimation accuracy and motion stability using the PRISM dataset. We evaluated ten conditions: five IMU configurations (2–6 IMUs), each tested with and without insole pressure sensors. IMUs were incrementally attached to both feet, both wrists (left only for the 3-IMU setup), the pelvis, and the head. MPJPE and PA-MPJPE were used for kinematic accuracy, and the Success Rate measured the proportion of sequences completed without falling. The quantitative results in Table 6.4 show that accuracy improves with more IMUs and further increases when foot pressure information is added. These findings indicate that increasing the number of IMUs enhances motion estimation accuracy, while foot pressure serves as a complementary modality that consistently improves both pose estimation and stability.

Compared with existing methods under the same sensor configuration in Table 6.2, the proposed method using two IMUs and insole pressure sensors shows lower performance than the baseline method FoRM. This result suggests that, in the physics simulation-based approach adopted in this study, observations from the feet alone are insufficient to stably drive the humanoid. While foot pressure information is effective for estimating contact states and load transitions, additional observations including the upper limbs are required to control full-body posture and balance within a physics simulator. Indeed, when IMUs are placed on four locations (both wrists and both feet) the control becomes more stable, enabling sufficiently accurate full-body motion estimation.

Furthermore, the primary reason why the state-of-the-art method GlobalPose, which uses six IMUs, achieves high local pose accuracy in Table 6.2 is that it constructs a root-relative coordinate system anchored at a pelvis-mounted IMU and applies posture corrections that account for non-inertial

effects induced by body rotations as well as gravity direction. In contrast, GRIP prioritizes practical usability and places IMUs only on the wrists and feet, which prevents applying this pelvis-IMU-based correction and explains the gap in local pose accuracy. For a fair comparison, we did not apply this correction even in the six-IMU condition in Table 6.4. Moreover, the decrease in Success Rate when using more than four IMUs, as observed in Table 6.4, is likely caused by sensor noise and velocity estimation errors from pelvis or head IMUs, which affect trunk motion estimation and destabilize balance control.

Table 6.4: Comparison across different sensor configurations.

| #IMU | Press. | MPJPE<br>[mm] (↓) | PA-MPJPE<br>[mm] (↓) | Succ. Rate<br>[%] (↑) |
|------|--------|-------------------|----------------------|-----------------------|
| 2    |        | 262.01            | 84.47                | 90.35                 |
| 3    |        | 209.27            | 57.30                | 88.98                 |
| 4    |        | 194.48            | 49.97                | 88.58                 |
| 5    |        | 186.95            | 45.68                | 86.61                 |
| 6    |        | <u>160.76</u>     | <u>41.44</u>         | 87.01                 |
| 2    | ✓      | 247.12            | 82.36                | <u>93.90</u>          |
| 3    | ✓      | 199.60            | 56.80                | 91.14                 |
| 4    | ✓      | 182.44            | 46.47                | <b>94.49</b>          |
| 5    | ✓      | 164.25            | 42.82                | 90.94                 |
| 6    | ✓      | <b>143.06</b>     | <b>39.13</b>         | 92.32                 |

**Height Map Analysis.** This experiment aims to evaluate the contribution of the height map to motion estimation accuracy and stability. DynamicsNet incorporates a height map to represent the geometric structure of the environment, encoding the shapes of the ground and object surfaces around the humanoid. Using the PRISM dataset, we conduct a comparative experiment with and without the height map to assess its impact on motion estimation performance.

Table 6.5 shows the quantitative results. Using the height map reduces

both MPJPE and PA-MPJPE and improves the task success rate, indicating more accurate and stable motion estimation. The evaluated scenarios consist only of flat ground or flat ground with objects, without terrain variations such as slopes or steps. Since the height map over flat ground takes a uniform value (zero), it cannot contribute to improving estimation accuracy for flat-ground locomotion alone. Therefore, the observed improvements can be attributed to motions involving contact and interaction with objects. Notably, although no explicit mechanism is introduced to directly correct the distance between the humanoid and objects, incorporating the height map allows the humanoid to better capture object surface geometry, leading to improved posture stability and contact consistency during object interactions.

Table 6.5: Comparison with and without the height map.

| Height Map | MPJPE  | PA-MPJPE | Succ. Rate |
|------------|--------|----------|------------|
| w/         | 182.44 | 46.47    | 94.49      |
| w/o        | 189.11 | 48.57    | 93.31      |

**Ablation Study.** This experiment analyzes the contribution of kinematic information estimated by KinematicsNet to the control performance of DynamicsNet. In our framework, DynamicsNet, which drives a humanoid in a physics simulator, is an essential core module and is therefore kept fixed across all settings. We perform an ablation by progressively enabling kinematic information provided by KinematicsNet and its submodules within the State Difference observation, thereby evaluating their individual and collective contributions. Across all configurations, DynamicsNet receives the same sensor measurements  $\mathbf{O}_t^{\text{sen}}$  (including IMU and foot pressure signals), self observations  $\mathbf{O}_t^{\text{self}}$ , and environment observations  $\mathbf{O}_t^{\text{env}}$ . The ablation is conducted by incrementally augmenting kinematic information as follows:

Table 6.6: Ablation study on kinematic estimation used in DynamicsNet.

| Configuration                                           | MPJPE<br>[mm] ( $\downarrow$ ) | PA-MPJPE<br>[mm] ( $\downarrow$ ) | Succ. Rate<br>[%] ( $\uparrow$ ) |
|---------------------------------------------------------|--------------------------------|-----------------------------------|----------------------------------|
| <b>Sensors</b>                                          | 290.71                         | 51.83                             | 90.89                            |
| <b>Sensors + KinNet(<math>v</math>)</b>                 | 206.50                         | 48.97                             | 91.25                            |
| <b>Sensors + KinNet(<math>v, p_{\text{glo}}</math>)</b> | 187.86                         | 48.87                             | 93.31                            |
| <b>Sensors + KinNet(<math>v, p_{\text{rel}}</math>)</b> | <b>182.44</b>                  | <b>46.47</b>                      | <b>94.49</b>                     |

- **Sensors:** DynamicsNet is driven only by raw sensor measurements, without any kinematic estimates.
- **Sensors + KinNet( $v$ ):** Estimated leaf-joint velocities are provided, corresponding to enabling the local velocity estimation submodule.
- **Sensors + KinNet( $v, p_{\text{glo}}$ ):** Globally reconstructed joint positions are additionally incorporated, corresponding to enabling pose estimation submodules that integrate root velocities.
- **Sensors + KinNet( $v, p_{\text{rel}}$ ):** Root-relative joint position estimates are used instead of global joint positions, avoiding accumulated integration errors and corresponding to the full KinematicsNet configuration.

We evaluate all configurations using MPJPE, PA-MPJPE, and Success Rate. The quantitative results are summarized in Table 6.6. The results show that progressively enabling kinematic estimation within KinematicsNet improves both kinematic accuracy and control stability, demonstrating that KinematicsNet plays a crucial role in stabilizing physics-based humanoid control.

**Performance Analysis.** This section provides a more detailed analysis of the performance of the proposed method. As shown in Fig. 6.11, the success rates across different motion categories indicate high performance for



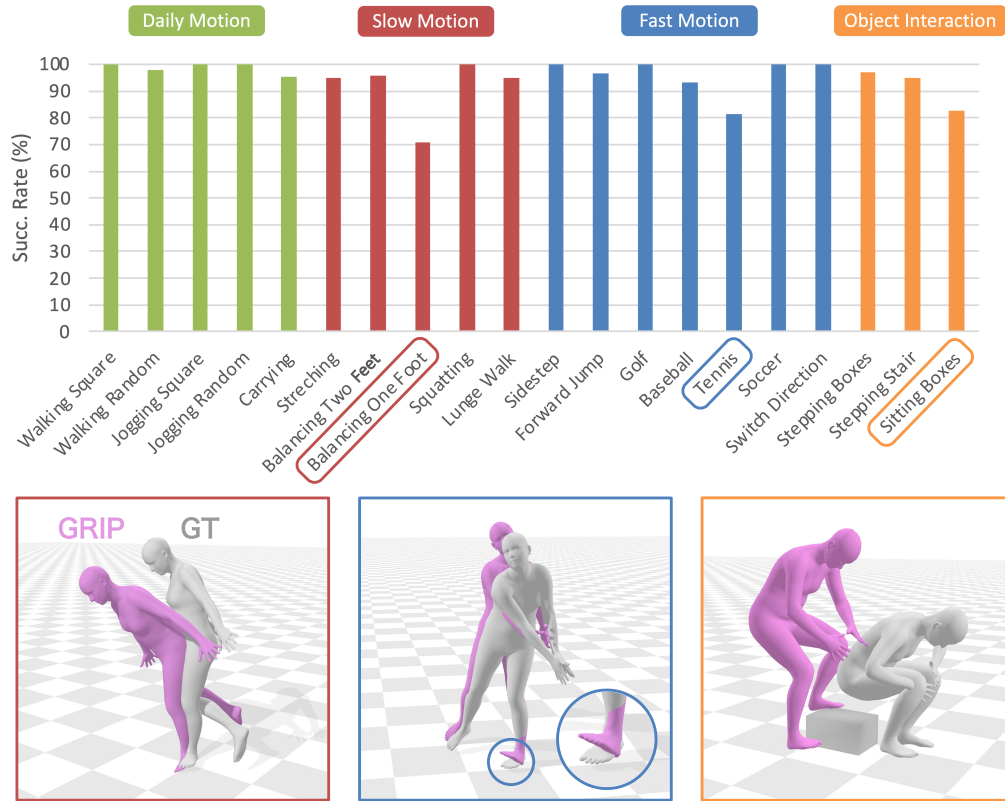


Figure 6.11: Success rates for different motion categories. Lower success rates are observed for balance motions, fast movements, and object interaction tasks compared to daily motions.

daily motions and relatively slow movements, while lower success rates are observed for balance-related motions such as Balancing One Foot, fast movements such as tennis, and motions involving object interactions. These failures can be attributed to the difficulty of reproducing dynamic and unstable motions involving the upper body, as well as to foot-ground friction effects during motion and failures in object interactions caused by accumulated position errors.

Next, we analyze the impact of sequence length on estimation accuracy. As shown in Table 6.7, while PA-MPJPE remains nearly unchanged when

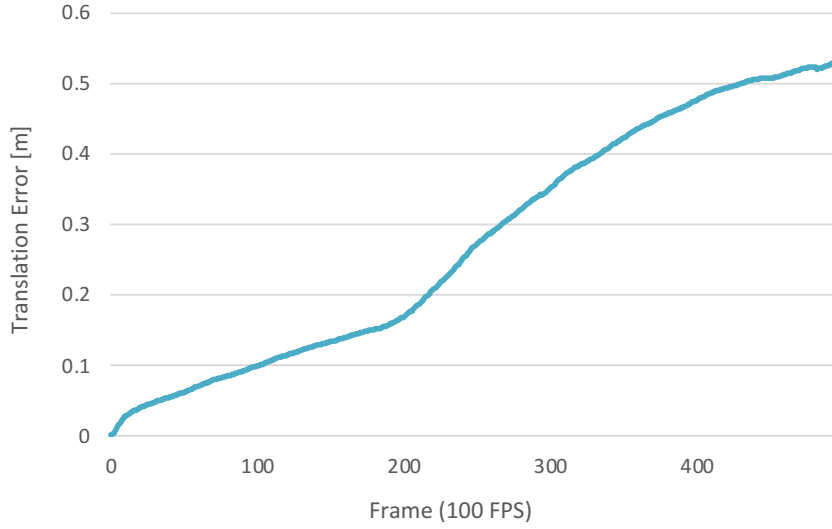


Figure 6.12: Temporal evolution of translation error over a sequence. The translation error increases approximately linearly with time, reflecting the accumulation of drift in absolute position estimation.

Table 6.7: Comparison on different sequence lengths.

| Seq. Length | MPJPE  | PA-MPJPE | Succ. Rate |
|-------------|--------|----------|------------|
| 5 sec       | 182.44 | 46.47    | 94.49      |
| 10 sec      | 238.61 | 46.59    | 85.83      |

comparing 5-second and 10-second sequences, both MPJPE and success rate degrade for the longer sequences. This indicates that the estimation of joint structure itself remains relatively stable, whereas absolute position errors accumulate over time due to drift. The temporal evolution of translation error in Fig. 6.12 further confirms that position error increases approximately linearly with time.

Finally, we discuss the computational cost of the proposed method. Training requires approximately 24 hours for KinematicsNet and 48 hours for DynamicsNet, while inference runs at approximately 1.5 ms and 26.5

ms per frame, respectively, which is sufficient for real-time operation. However, the physical simulation remains a primary computational bottleneck. Moreover, considering additional processing required for online operation, such as data transfer and real-time visualization, which are not supported in the current system, further improvements in inference efficiency will be important for realizing future real-time demonstration systems.

## 6.4 Summary

We presented GRIP, a physics-based human motion capture framework that integrates four wearable IMUs, insole foot pressure sensors, and physics simulation, and introduced PRISM, a new multimodal dataset capturing diverse human motions and physical interactions. GRIP leverages pressure-based dynamic cues and simulation-based physical constraints to achieve accurate global trajectory estimation and physically plausible pose estimation with a minimal sensor setup. Future directions include designing controllers that can better estimate stable motions under unstable or high-momentum conditions, integrating additional sensing modalities such as cameras or localization sensors to further reduce drift, and extending GRIP to multi-person scenarios and dynamic object interactions for broader applicability in robotics, VR/AR, and biomechanics.

# Chapter 7

## Conclusion

### 7.1 Summary

This thesis aims to realize learning-based human motion capture that functions reliably in everyday environments, addressing three fundamental challenges: (1) the vulnerability of vision sensors to environmental factors, (2) the burden associated with wearing multiple sensing devices, and (3) the physical inconsistencies that often arise in data-driven motion estimation. To tackle these issues, we presented several approaches that utilize multimodal sensors and physics-based modeling.

Chapter 3 introduced an egocentric 3D pose estimation method using a wrist-mounted 360-degree camera, offering a solution to the field-of-view limitations and occlusions inherent to external camera setups. Chapter 4 proposed an approach for reconstructing human meshes solely from event data, improving robustness to illumination changes by eliminating the dependency on conventional frame-based imagery. Chapter 5 presented a gait estimation method using only shoe-embedded IMUs, demonstrating that practical accuracy can be achieved while keeping the wearable load minimal. Chapter 6 proposed a method that integrates smartwatches and

insole pressure sensors with physics simulation, enabling the reconstruction of dynamically consistent full-body motion from a small set of sensors.

These four approaches each address different aspects of the problem, yet they possess complementary strengths. Vision sensors enable accurate position and pose estimation based on visual cues, but their performance degrades under occlusions caused by objects or the user's own body. In contrast, IMU sensors are unaffected by visual occlusions and can provide stable 3D motion estimates, thereby complementing vision-based methods. Furthermore, foot-pressure data and physics-based modeling help ensure consistency in body dynamics and interactions with the environment, adding physical plausibility to the reconstructed motions. Taken together, these characteristics suggest that combining modalities appropriately, depending on the application scenario and environmental conditions, can enable more reliable motion capture systems.

The outcomes of this research serve as an important foundation for the development of flexible and practical motion-sensing technologies that can operate robustly within everyday environments.

## 7.2 Limitations

The studies presented in this thesis demonstrate that the proposed approaches can address several challenges of learning-based human motion capture and expand their applicability to everyday environments, but several limitations remain when considering practical deployment.

First, many of the methods developed in this thesis assume constrained settings. Each approach primarily targets single-person motion and does not sufficiently address more complex scenarios such as multi-person activities, human-to-human interactions, or interactions with moving objects. In addition, they generally assume contact reasoning with static environ-

ments, which limits their ability to handle more dynamic interaction settings. Second, the methods introduced in each chapter were designed independently for their respective sensing modalities and are not yet integrated into a unified framework, making it difficult to achieve flexible multimodal fusion or dynamic adaptation to diverse user environments. Third, the proposed approaches rely on datasets and body models collected mainly from healthy individuals, making it challenging to accommodate the diversity of physical characteristics or patient-specific conditions relevant to medical or rehabilitation settings. This leads to limitations in applications that require high-precision reconstruction of individualized anatomical structures.

Addressing these issues will be essential for establishing motion sensing technologies that operate more flexibly and reliably in real-world environments.

### 7.3 Discussion and Future Work

Human digital twins and next-generation robotics have the potential to support a wide range of applications, including industrial operations, healthcare and rehabilitation, sports analysis, education, and smart workplaces. Realizing such systems requires not only accurate motion estimation, but also an integrated design of sensing, modeling, and system-level components that can operate robustly in everyday environments.

Building on the methods proposed in this thesis, further extensions are required from both research and system design perspectives. Accordingly, this section is organized into two parts. We first discuss future research directions that advance the technical foundations of motion capture, and then outline system design perspectives toward practical deployment in real-world settings.

### 7.3.1 Research Directions

Future research directions for human motion capture and digital twin systems can be organized around three dimensions: sensing, modeling, and understanding, which together form their technical foundation.

**Sensing.** Future sensing technologies should capture human motion and environmental states across diverse scenarios. As wearable devices such as smart glasses and egocentric vision systems become more common, integrating first-person cameras with IMUs and insole sensors will enable richer and more flexible motion capture. Such multimodal sensing will be particularly valuable for fine-grained indoor activities, large-scale environments, and motions involving interactions with dynamic objects, as well as scenarios in which multiple people are present.

**Modeling.** From a modeling perspective, an important challenge is to construct unified frameworks that integrate multiple sensor modalities and account for interactions at different scales. Models that jointly represent Human–Environment, Human–Object, and Human–Human interactions can provide more adaptable mechanisms to reconstruct real-world behaviors. Toward this goal, human motion foundation models that can adapt to varying tasks and environments are a promising direction, with potential benefits for both motion capture and humanoid robot control.

**Understanding.** Beyond motion reconstruction, understanding of human behavior and internal physical states remains an important research direction. Estimating physical workload and task-related risk can contribute to safety monitoring and injury prevention, while quantitative analysis of individual differences supports personalized assessment in healthcare and movement science. In human–robot collaboration, motion-based understanding will also be essential for intent inference and motion anticipation.

### 7.3.2 System Design Perspectives

From a system design perspective, deploying human motion capture technologies in real-world settings requires careful consideration beyond algorithmic performance, including application requirements, sensing configurations, computational constraints, social acceptance, and the evaluation of social impact.

**Application-Driven Requirements Definition.** Practical deployment of human motion capture systems requires defining required functions and accuracy levels according to the target application. These requirements vary widely across domains such as biomechanics, healthcare, industrial safety, and sports analysis. For instance, biomechanical analysis often targets joint angle errors below approximately 5 degrees, whereas safety monitoring may only require coarse posture recognition or anomaly detection. Therefore, performance evaluation based solely on comparisons with prior methods is insufficient to assess real-world suitability. In addition, applications with real-time or near-real-time constraints require balancing estimation accuracy with latency. Factors such as long-term stability, wearing burden, and robustness further influence deployability, highlighting the importance of application-driven system design.

**Sensor Selection and Modality Trade-offs.** The selection of sensing modalities directly affects estimation accuracy, environmental robustness, and deployment feasibility. Cameras provide rich geometric and environmental information but are constrained by installation conditions and privacy concerns. IMUs enable environment-independent motion capture suitable for daily use, though they suffer from drift in absolute estimation. Insole pressure sensors offer direct measurements of contact states and load transitions, improving physical consistency, but are limited to lower-body interactions. As these modalities offer complementary strengths and weak-



nesses, no single sensor can satisfy all requirements. Practical motion capture systems therefore require careful sensor selection and combination tailored to the target application and environment.

**Method Selection and Computational Trade-offs.** Method selection significantly influences both estimation quality and system practicality. Kinematic approaches are generally efficient and suitable for real-time processing but have limited capability to model physical interactions. Physics-based methods can enforce contact and dynamic consistency, improving stability, at the cost of higher computational demands. Model architecture choices further affect latency and resource usage, making lightweight models preferable in some scenarios. These trade-offs are closely linked to deployment settings, such as cloud-based processing, local workstations, or edge devices. Effective system design therefore requires balancing accuracy, computational efficiency, real-time capability, and robustness.

**Social Acceptance and Ethical Considerations.** Beyond technical performance, social acceptance plays a critical role in real-world deployment. Motion capture in everyday environments may raise privacy concerns, but many technologies have been socially accepted when their benefits clearly outweigh perceived risks and their usage conditions are well defined. This suggests that deployment depends not only on minimizing privacy risks, but also on designing and communicating appropriate trade-offs. Technical choices such as avoiding appearance capture, relying on numerical representations, and enabling on-device processing can support this process. In domains such as industrial and healthcare settings, where benefits and responsibilities are explicit, consensus building is often more feasible. Overall, successful deployment requires aligning technical design with application context and societal expectations.

**Measuring Social Impact.** In the social deployment of motion capture

technologies, user scale alone is not a sufficient measure of impact. Unlike speech or language applications that readily achieve large-scale adoption, body-mounted motion capture systems face inherent barriers to mass deployment, while still offering strong potential to address societal challenges such as workforce shortages and increasing care demands. As this research is further developed, the focus should therefore be placed not on maximizing user numbers, but on demonstrating practical value through sustained use, user and practitioner feedback, and measurable improvements in real-world workflows. Application development should focus on iteratively validating, operating, and improving the system in real-world settings, rather than on large-scale user acquisition. Demonstrating frequent, long-term use and concrete on-site benefits will be essential for clarifying the social significance of the proposed approach.

# Bibliography

- [1] Sadegh Aliakbarian, Pashmina Cameron, Federica Bogo, Andrew Fitzgibbon, and Thomas J. Cashman. FLAG: Flow-based 3D avatar generation from sparse observations. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13243–13252, 2022.
- [2] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A. S. Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A. Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H. Al-Timemy, Ye Duan, Amjed Abdullah, Laith Farhan, Yi Lu, Ashish Gupta, Felix Albu, Amin Abbosh, and Yuantong Gu. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10(46), 2023.
- [3] Sizhe An and Umit Y. Ogras. Fast and scalable human pose estimation using mmwave point cloud. In *ACM/IEEE Des. Autom. Conf.*, page 889–894, 2022.
- [4] Rokoko Electronics ApS. Rokoko - smart suits, motion capture, and virtual production. URL: <https://www.rokoko.com/>, Accessed: December 3, 2025.
- [5] Rayan Armani, Changlin Qian, Jiayi Jiang, and Christian Holz. Ultra inertial poser: Scalable motion capture and tracking from sparse inertial sensors and ultra-wideband ranging. In *ACM SIGGRAPH*

- Conf. Proc. (SIGGRAPH)*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [6] R. Wes Baldwin, Ruixu Liu, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Time-ordered recent event (TORE) volumes for event cameras. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 45(2):2519–2532, 2023.
- [7] Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, and C.V. Jawahar. Unsupervised learning of deep feature representation for clustering egocentric actions. In *Int. Jt. Conf. Artif. Intell. (IJCAI)*, pages 1447–1453, 2017.
- [8] Mirjam Bonanno, Alessandro M. De Nunzio, Angelo Quartarone, Angelo Militi, Francesco Petralito, and Rocco Salvatore Calabrò. Gait analysis in neurorehabilitation: From research to clinical practice. *Bioengineering (Basel)*, 10(7):785, Jun 2023.
- [9] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A  $240 \times 180$  130 db  $3 \mu\text{s}$  latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10):2333–2341, 2014.
- [10] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck. DHP19: Dynamic vision sensor 3D human pose dataset. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. Workshops (CVPRW)*, pages 1695–1704, 2019.
- [11] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 43(1):172–186, 2021.
- [12] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2019.

- [13] Lena Carcreff, Corinna N. Gerber, Anisoara Paraschiv-Ionescu, Gerardo De Coulon, Christopher J. Newman, Kamiar Aminian, and Stéphane Armand. Comparison of gait characteristics between clinical and daily life settings in children with cerebral palsy. *Scientific Reports*, 10:2091, 2020.
- [14] Angela Castillo, Maria Escobar, Guillaume Jeannerete, Albert Pumarola, Pablo Arbeláez, Ali Thabet, and Artsiom Sanakoyeu. BoDiffusion: Diffusing sparse observations for full-body human motion synthesis. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.
- [15] Young-Woon Cha, True Price, Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, Zihe Qin, Hyounghun Kim, Zhaoqi Su, Yebin Liu, et al. Towards fully mobile 3D face, body, and environment capture using only head-worn cameras. *IEEE Trans. Vis. Comput. Graphics (TVCG)*, 24(11):2993–3004, 2018.
- [16] Young-Woon Cha, Husam Shaik, Qian Zhang, Fan Feng, Andrei State, Adrian Ilie, and Henry Fuchs. Mobile. egocentric human body motion reconstruction using only eyeglasses-mounted cameras and a few body-worn inertial sensors. In *IEEE Conf. Virtual Real. 3D User Interfaces (VR)*, pages 616–625, 2021.
- [17] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 77–85, 2017.
- [18] Jiaan Chen, Hao Shi, Yaozu Ye, Kailun Yang, Lei Sun, and Kaiwei Wang. Efficient human pose estimation via 3D event point cloud. In *Int. Conf. 3D Vis. (3DV)*, pages 1–10, 2022.
- [19] Wenqiang Chen, Yexin Hu, Wei Song, Yingcheng Liu, Antonio Torralba, and Wojciech Matusik. CAAvatar: Real-time human activity mesh reconstruction via tactile carpets. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, 7(4), 2024.

- [20] Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Comput. Vis. Image Underst.*, 192:102897, 2020.
- [21] Sangbum Choi, Seokeon Choi, and Changick Kim. MobileHuman-Pose: Toward real-time 3d human pose estimation in mobile devices. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pages 2328–2338, Jun 2021.
- [22] Peng Dai, Yang Zhang, Tao Liu, Zhen Fan, Tianyuan Du, Zhuo Su, Xiaozheng Zheng, and Zeming Li. HMD-Poser: On-device real-time human motion tracking from scalable sparse observations. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.
- [23] Inc. DARI Motion. Captury. URL: <https://captury.com/>, Accessed: December 3, 2025.
- [24] J. Deng, W. Dong, R. Socher, L.-J Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 248–255, 2009.
- [25] Andrea Dittadi, Sebastian Dziadzio, Darren Cosker, Ben Lundell, Thomas J. Cashman, and Jamie Shotton. Full-body motion from a single head-mounted device: Generating SMPL poses from partial observations. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 11687–11697, 2021.
- [26] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars Grow Legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023.
- [27] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2353–2362, 2017.

- [28] Alireza Fathi, Ali Farhadi, and James M. Rehg. Understanding ego-centric activities. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 407–414, 2011.
- [29] Han Feng, Wenchao Ma, Quankai Gao, Xianwei Zheng, Nan Xue, and Huijuan Xu. SAGE: Stratified avatar generation from sparse observations. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024.
- [30] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In *Conf. Robot Learn. (CoRL)*, pages 138–149, 2022.
- [31] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 44(1):154–180, 2022.
- [32] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.
- [33] Gaurvi Goyal, Franco Di Pietro, Nicolo Carissimi, Arren Glover, and Chiara Bartolozzi. MoveEnet: Online high-frequency human pose estimation with an event camera. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. Workshops (CVPRW)*, pages 4024–4033, 2023.
- [34] Vladimir Guzov, Julian Chibane, Riccardo Marin, Yannan He, Yunus Saracoglu, Torsten Sattler, and Gerard Pons-Moll. Interaction replica: Tracking human–object interaction and scene changes from human motion. In *International Conference on 3D Vision (3DV)*, March 2024.
- [35] Vladimir Guzov, Yifeng Jiang, Fangzhou Hong, Gerard Pons-Moll, Richard Newcombe, C. Karen Liu, Yuting Ye, and Lingni Ma. Hmd<sup>2</sup>: Environment-aware motion generation from a single egocentric head-mounted device. In *Proc. Int. Conf. 3D Vision (3DV)*, Mar 2025.

- [36] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human POSEitioning system (HPS): 3D human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, jun 2021.
- [37] Erik Gärtner, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu. Differentiable dynamics for articulated 3d human motion reconstruction. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13180–13190, 2022.
- [38] Erik Gärtner, Mykhaylo Andriluka, Hongyi Xu, and Cristian Sminchisescu. Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 13096–13105, 2022.
- [39] Xingjian Han, Ben Senderling, Stanley To, Deepak Kumar, Emily Whiting, and Jun Saito. GroundLink: A dataset unifying human body movement and ground reaction dynamics. In *ACM SIGGRAPH Asia Conf. Papers (SA)*, 2023.
- [40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 770–778, 2016.
- [41] Koyu Hori, Yufeng Mao, Yumi Ono, Hiroki Ora, Yuki Hirobe, Hiroyuki Sawada, Akira Inaba, Satoshi Orimo, and Yoshihiro Miyake. Inertial measurement unit-based estimation of foot trajectory for clinical gait analysis. *Frontiers in Physiology*, 10, 2020.
- [42] Ryosuke Hori, Hiroyuki Deguchi, Tsubasa Maruyama, Mitsunori Tada, and Hideo Saito. Gait inertial poser (GIP): Gait-aware human motion capture using shoe-embedded IMUs. *IEEE Access*, 13:183262–183282, 2025.
- [43] Ryosuke Hori, Ryo Hachiuma, Hideo Saito, Mariko Isogawa, and Dan Mikami. Silhouette-based synthetic data generation for 3d hu-



- man pose estimation with a single wrist-mounted 360° camera. In *Proc. Int. Conf. Image Process. (ICIP)*, pages 1304–1308, 2021.
- [44] Tao Hu, Kripasindhu Sarkar, Lingjie Liu, Matthias Zwicker, and Christian Theobalt. EgoRenderer: Rendering human avatars from egocentric camera images. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 14528–14538, Oct 2021.
- [45] Buzhen Huang, Liang Pan, Yuan Yang, Jingyi Ju, and Yangang Wang. Neural MoCon: Neural Motion Control for Physically Plausible Human Motion Capture . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6407–6416, Los Alamitos, CA, USA, June 2022. IEEE Computer Society.
- [46] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser (DIP): Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Trans. Graph. (TOG)*, 37(6), 2018.
- [47] Yonatan Hutabarat, Dai Owaki, and Mitsuhiro Hayashibe. Recent advances in quantitative gait analysis using wearable sensors: A review. *IEEE Sensors Journal*, 21(23):26470–26487, Dec 2021.
- [48] Dae-Hyeok Hwang, Kazuki Aso, Ye Yuan, Kris Kitani, and Hideki Koike. MonoEye: Multimodal human motion capture system using a single ultra-wide fisheye camera. In *Proc. ACM Symp. User Interface Softw. Technol. (UIST)*, pages 98–111, 2020.
- [49] Movella Inc. Xsens imu motion capture sensor. URL: <https://www.movella.com/products/xsens>, Accessed: December 3, 2025.
- [50] Theia Markerless Inc. Theia3d markerless motion capture systems. URL: <https://www.theiamarkerless.ca/>, Accessed: December 3, 2025.

- [51] M. Isogawa, Y. Yuan, M. O'Toole, and K. Kitani. Optical non-line-of-sight physics-based 3D human pose estimation. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7011–7020, 2020.
- [52] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5967–5976, 2017.
- [53] H. Jiang and K. Grauman. Seeing invisible poses: Estimating 3D body pose from egocentric video. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3501–3509, 2017.
- [54] Hao Jiang and Vamsi Krishna Ithapu. Egocentric pose estimation from human vision span. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 11006–11014, Oct 2021.
- [55] Jiayi Jiang, Paul Streli, Xuejing Luo, Christoph Gebhardt, and Christian Holz. MANIKIN: Biomechanically accurate neural inverse kinematics for human motion estimation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.
- [56] Jiayi Jiang, Paul Streli, Manuel Meier, and Christian Holz. EgoPoser: Robust real-time egocentric pose estimation from sparse and intermittent observations everywhere. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.
- [57] Jiayi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. AvatarPoser: Articulated full-body pose tracking from sparse motion sensing. In *Eur. Conf. Comput. Vis. (ECCV)*, 2022.
- [58] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. Towards 3D human pose construction using wifi. In *Int. Conf. Mob. Comput. Netw., MobiCom*, 2020.

- [59] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W. Winkler, and C. Karen Liu. Transformer inertial poser: Real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *ACM SIGGRAPH Asia Conf. Papers (SA)*, 2022.
- [60] Jingyi Ju, Buzhen Huang, Chen Zhu, Zhihao Li, and Yangang Wang. Physics-guided human motion capture with pose probability modeling. In *Int. Jt. Conf. Artif. Intell. (IJCAI)*, IJCAI '23, 2023.
- [61] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 7122–7131, 2018.
- [62] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik. Learning 3D human dynamics from video. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 5607–5616, 2019.
- [63] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent. (ICLR)*, 2015.
- [64] Yoshiyuki Kobayashi, Naoto Hida, Kanako Nakajima, Masahiro Fujimoto, and Masaaki Mochimaru. Aist gait database 2019, 2019.
- [65] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video inference for human body pose and shape estimation. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 5252–5262, 2020.
- [66] Yuji Kumano, Suguru Kanoga, Masataka Yamamoto, Hiroshi Take-mura, and Mitsunori Tada. Estimating whole-body walking motion from inertial measurement units at wrist and heels using deep learning. *International Journal of Automation Technology*, 17(3):217–225, 2023.
- [67] Jogendra Nath Kundu, Nishank Lakkakula, and Venkatesh Babu Radhakrishnan. UM-Adapt: Unsupervised multi-task adaptation

- using adversarial cross-task distillation. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 1436–1445, 2019.
- [68] Daniel Laidig, Andreas J. Jocham, Bernhard Guggenberger, Klemens Adamer, Michael Fischer, and Thomas Seel. Calibration-free gait assessment by foot-worn inertial sensors. *Frontiers in Digital Health*, 3, 2021.
- [69] Daniel Laidig and Thomas Seel. VQF: Highly accurate IMU orientation estimation with bias estimation and magnetic disturbance rejection. *Inf. Fusion*, 91:187–204, 2023.
- [70] Jiye Lee and Hanbyul Joo. Mocap everyone everywhere: Lightweight motion capture with smartwatches and a head-mounted camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [71] Sunmin Lee, Sebastian Starke, Yuting Ye, Jungdam Won, and Alexander Winkler. QuestEnvSim: Environment-aware simulated motion tracking from sparse sensors. In *ACM SIGGRAPH Conf. Proc. (SIGGRAPH)*, 2023.
- [72] Jiefeng Li, Siyuan Bian, Chao Xu, Gang Liu, Gang Yu, and Cewu Lu. D&D: Learning human dynamics from dynamic camera. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, page 479–496, Berlin, Heidelberg, 2022. Springer-Verlag.
- [73] Tianhong Li, Lijie Fan, Mingmin Zhao, Yingcheng Liu, and Dina Katabi. Making the invisible visible: Action recognition through walls and occlusions. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 872–881, 2019.
- [74] Yanjie Li, Sen Yang, Peidong Liu, Shoukui Zhang, Yunxiao Wang, Zhicheng Wang, Wankou Yang, and Shu-Tao Xia. SimCC: A simple coordinate classification perspective for human pose estimation. In *Eur. Conf. Comput. Vis. (ECCV)*, page 89–106, 2022.

- [75] Huakun Liu, Hiroki Ota, Xin Wei, Yutaro Hirao, Monica Perusquía-Hernández, Hideaki Uchiyama, and Kiyoshi Kiyokawa. UMotion: Uncertainty-driven human motion estimation from inertial and ultra-wideband units. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7085–7094, 2025.
- [76] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 10012–10022, Oct 2021.
- [77] Z. Liu, J. Zhu, J. Bu, and C. Chen. A survey of human pose estimation: The body parts parsing based methods. *J. Vis. Commun. Image Represent.*, 32:10–19, 2015.
- [78] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Int. Conf. Mach. Learn. (ICML)*, ICML’15, page 97–105. JMLR.org, 2015.
- [79] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph. (TOG)*, 34(6), 2015.
- [80] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM SIGGRAPH Asia Conf. Papers (SA)*, 33(6):220:1–220:13, 2014.
- [81] Noitom Ltd. Noitom imu motion capture systems. URL: <https://www.noitom.com/>, Accessed: December 3, 2025.
- [82] Vicon Motion Systems Ltd. Vicon. URL: <https://www.vicon.com/>, Accessed: December 3, 2025.
- [83] Yiyue Luo, Yunzhu Li, Michael Foshey, Wan Shou, Pratyusha Sharma, Tomás Palacios, Antonio Torralba, and Wojciech Matusik. Intelligent carpet: Inferring 3D human pose from tactile signals. In

- IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 11250–11260, 2021.
- [84] Zhengyi Luo, Jinkun Cao, Rawal Khirodkar, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Real-Time Simulated Avatar from Head-Mounted Sensors . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 571–581, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.
- [85] Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [86] Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10861–10870, 2023.
- [87] Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. Dynamics-regulated kinematic policy for egocentric pose estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [88] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2022.
- [89] Xiaolei Lv, Jinxiang Chai, and Shihong Xia. Data-driven inverse dynamics for human motion. *ACM Trans. Graph. (TOG)*, 35(6), Dec. 2016.
- [90] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael Black. AMASS: Archive of motion capture as surface shapes. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 5441–5450, 2019.

- [91] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2659–2668, 2017.
- [92] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *Proc. Int. Conf. 3D Vis. (3DV)*, pages 506–516, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.
- [93] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D human pose estimation with a single rgb camera. *ACM Trans. Graph. (TOG)*, 36(4), 2017.
- [94] MMSegmentation Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mms Segmentation>, 2020.
- [95] Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. IMUPoser: Full-body pose estimation using imus in phones, watches, and earbuds. In *ACM Conf. Hum. Factors Comput. Syst. (CHI)*, 2023.
- [96] Lucas Mourot, Ludovic Hoyet, François Le Clerc, and Pierre Hellier. UnderPressure: Deep learning for foot contact detection, ground reaction force estimation and footskate cleanup. *Comput. Graph. Forum*, 41(8):195–206, 2022.
- [97] Marion Mundt, Arnd Koeppel, Franz Bamer, Sina David, and Bernd Markert. Artificial neural networks in motion analysis—applications of unsupervised and heuristic feature selection techniques. *IEEE Sensors Journal*, 20(16):4581, 2020.
- [98] Marion Mundt, Arnd Koeppel, Sina David, Tom Witter, Franz Bamer, Wolfgang Potthast, and Bernd Markert. Estimation of gait mechanics

- based on simulated and measured imu data using an artificial neural network. *Frontiers in Bioengineering and Biotechnology*, 8:41, 2020.
- [99] Inc NaturalPoint. OptiTrack. URL: <https://optitrack.com/>, Accessed: December 3, 2025.
- [100] E. Ng, D. Xiang, H. Joo, and K. Grauman. You2Me: Inferring body pose in egocentric video via first and second person interactions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 9887–9897, 2020.
- [101] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 1263–1272, 2017.
- [102] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3D human pose annotations. In *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 1253–1262, 2017.
- [103] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph. (TOG)*, 40(4), 2021.
- [104] Mugalodi Rakesh, Jogendra Nath Kundu, Varun Jampani, and Venkatesh Babu Radhakrishnan. Aligning silhouette topology for self-adaptive 3D human pose recovery. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021.
- [105] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.
- [106] Shenghao Ren, Yi Lu, Jiayi Huang, Jiayi Zhao, He Zhang, Tao Yu, Qiu Shen, and Xun Cao. MotionPRO: Exploring the role of pressure in human mocap and beyond. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 27760–27770, 2025.



- [107] Yili Ren, Zi Wang, Yichao Wang, Sheng Tan, Yingying Chen, and Jie Yang. GoPose: 3D human pose estimation using wifi. *ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, 6(2), 2022.
- [108] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. EgoCap: Egocentric marker-less motion capture with two fisheye cameras. *ACM Trans. Graph. (TOG)*, 35(6), 2016.
- [109] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3D human pose estimation from multi-view images. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun 2018.
- [110] G. Rogez, J. S. Supančič, and D. Ramanan. First-person pose recognition using egocentric workspaces. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4325–4333, 2015.
- [111] Viktor Rudnev, Vladislav Golyanik, Jiayi Wang, Hans-Peter Seidel, Franziska Mueller, Mohamed Elgharib, and Christian Theobalt. EventHands: Real-time neural 3d hand pose estimation from an event stream. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 12365–12375, 2021.
- [112] Abdul Saboor, Triin Kask, Alar Kuusik, Muhammad Mahtab Alam, Yannick Le Moullec, Imran Khan Niazi, Ahmed Zoha, and Rizwan Ahmad. Latest research trends in gait analysis using wearable sensors and machine learning: A systematic review. *IEEE Access*, 8:167830–167864, 2020.
- [113] N. Sarafianos, B. Boteanu, B. Ionescu, and I.A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Comput. Vis. Image Underst.*, 152:1–20, 2016.

- [114] Gianluca Scarpellini, Pietro Morerio, and Alessio Del Bue. Lifting monocular events to 3D human poses. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. Workshops (CVPRW)*, pages 1358–1368, 2021.
- [115] Abigail C. Schmitt, Sidney T. Baudendistel, Ania L. Lipat, Tatiana A. White, Tiphane E. Raffegeau, and Chris J. Hass. Walking indoors, outdoors, and on a treadmill: Gait differences in healthy young and older adults. *Gait & Posture*, 90:468–474, Oct 2021. Epub 2021 Sep 28.
- [116] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. PPO: Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [117] Jesse Scott, Bharadwaj Ravichandran, Christopher Funk, Robert T. Collins, and Yanxi Liu. From image to stability: Learning dynamics from human pose. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 536–554, 2020.
- [118] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-Pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17):10032–10044, 2020.
- [119] V. V. Shah, J. McNames, M. Mancini, P. Carlson-Kuhta, Rebecca I. Spain, J. G. Nutt, M. El-Gohary, Carolin Curtze, and F. B. Horak. Laboratory versus daily life gait characteristics in patients with multiple sclerosis, parkinson’s disease, and matched controls. *Journal of NeuroEngineering and Rehabilitation*, 17:159, 2020.
- [120] Zhanpeng Shao, Xueping Wang, Wen Zhou, Wuzhen Wang, Jianyu Yang, and Youfu Li. A temporal densely connected recurrent network for event-based human pose estimation. *Pattern Recognition*, page 110048, 2023.
- [121] Mohsen Sharifi Renani, Abigail M. Eustace, Casey A. Myers, and Chadd W. Clary. The use of synthetic imu signals in the training of deep learning models significantly improves the accuracy of joint kinematic predictions. *IEEE Sensors Journal*, 21(17):5876, 2021.

- [122] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt. Neural monocular 3d human motion capture with physical awareness. *ACM Trans. Graph. (TOG)*, 40(4), July 2021.
- [123] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: physically plausible monocular 3d motion capture in real time. *ACM Trans. Graph. (TOG)*, 39(6), Nov. 2020.
- [124] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins. Motion capture from body-mounted cameras. *ACM Trans. Graph. (TOG)*, 30(4), Jul 2011.
- [125] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, page 8934–8943, 2018.
- [126] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, September 2018.
- [127] Unity Technologies. Unity. URL: <https://unity.com/>, Accessed: December 3, 2025.
- [128] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Trans. Graph. (TOG)*, 2024.
- [129] Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3D human mesh from monocular images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, pages 1–20, 2023.
- [130] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 5026–5033, 2012.
- [131] D. Tome, T. Alldieck, P. Peluse, G. Pons-Moll, L. Agapito, H. Badino, and F. De la Torre. SelfPose: 3D egocentric pose estimation from

- a headset mounted camera. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, page 1–1, 2020.
- [132] D. Tome, P. Peluse, L. Agapito, and H. Badino. xR-EgoPose: Ego-centric 3D human pose from an hmd camera. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 7727–7737, 2019.
- [133] Denis Tome, Chris Russell, and Lourdes de Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 5689–5698, 2017.
- [134] Shashank Tripathi, Lea Müller, Chun-Hao P. Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. 3D human pose estimation via intuitive physics. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4713–4725, 2023.
- [135] Hirotaka Uchitomi, Yuki Hirobe, and Yoshihiro Miyake. Three-dimensional continuous gait trajectory estimation using single shank-worn inertial measurement units and clinical walk test application. *Scientific Reports*, 12:5368, 2022.
- [136] Yuki Uno, Issei Ogasawara, Shoji Konda, Natsuki Yoshida, Naoki Otsuka, Yuya Kikukawa, Akira Tsujii, and Ken Nakata. Validity of spatio-temporal gait parameters in healthy young adults using a motion-sensor-based gait analysis system (orphe analytics) during walking and running. *IEEE Sensors Journal*, 23(1):331, 2023.
- [137] Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott Delp, and C. Karen Liu. Diffusionposer: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2513–2523, June 2024.
- [138] T. von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll. Sparse inertial poser (SIP): Automatic 3D human pose estimation from sparse IMUs. *Comput. Graph. Forum*, 36(2):349–360, 2017.

- [139] Jian Wang, Zhe Cao, Diogo Luvizon, Lingjie Liu, Kripasindhu Sarkar, Danhang Tang, Thabo Beeler, and Christian Theobalt. Ego-centric whole-body motion capture with fisheyevit and diffusion-based motion refinement. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 777–787, 2024.
- [140] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. Estimating egocentric 3D human pose in global space. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 11500–11509, Oct 2021.
- [141] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3D human pose estimation: A review. *Comput. Vis. Image Underst.*, 210:103225, 2021.
- [142] Alexander Winkler, Jungdam Won, and Yuting Ye. QuestSim: Human motion tracking from sparse sensors with simulated avatars. In *ACM SIGGRAPH Asia Conf. Papers (SA)*, 2022.
- [143] T. A. L. Wren, G. E. Gorton, S. Ounpuu, and C. A. Tucker. Efficacy of clinical gait analysis: A systematic review. *Gait & Posture*, 34(2):149–153, Jun 2011. Epub 2011 Jun 8.
- [144] T. A. L. Wren, C. A. Tucker, S. A. Rethlefsen, G. E. Gorton, and S. Öunpuu. Clinical efficacy of instrumented gait analysis: Systematic review 2020 update. *Gait & Posture*, 80:274–279, Jul 2020. Epub 2020 May 23.
- [145] Erwin Wu, Rawal Khirodkar, Hideki Koike, and Kris Kitani. SolePoser: Full-body pose estimation using a single pair of insole sensors. In *ACM Symp. User Interface Softw. Technol. (UIST)*, 2024.
- [146] Xinyu Xiao, Jun Wang, Peng Feng, et al. Fast human motion reconstruction from sparse inertial measurement units considering the human shape. *Nature Communications*, 15:2423, 2024.
- [147] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. Physics-based human motion estimation and

- synthesis from videos. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 11512–11521, 2021.
- [148] Lan Xu, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt. EventCap: Monocular 3D capture of high-speed human motions using an event camera. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 4967–4977, 2020.
- [149] Vasco Xu, Chenfeng Gao, Henry Hoffmann, and Karan Ahuja. MobilePoser: Real-time full-body pose estimation and 3d human translation from imus in mobile consumer devices. In *ACM Symp. User Interface Softw. Technol. (UIST)*, UIST '24, New York, NY, USA, 2024. Association for Computing Machinery.
- [150] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, P. Fua, H.-P. Seidel, and C. Theobalt. Mo2Cap2: Real-time mobile 3D motion capture with a cap-mounted fisheye camera. *IEEE Trans. Vis. Comput. Graphics (TVCG)*, 25(5):2093–2101, 2019.
- [151] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. MmMesh: Towards 3D real-time dynamic human mesh construction using millimeter-wave. In *ACM Int. Conf. Mobile Syst., Appl., Serv.*, page 269–282, 2021.
- [152] Brent Yi, Vickie Ye, Maya Zheng, Yunqi Li, Lea Müller, Georgios Pavlakos, Yi Ma, Jitendra Malik, and Angjoo Kanazawa. Estimating body and hand motion in an ego-sensed world. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7072–7084, 2025.
- [153] Xinyu Yi, Shaohua Pan, and Feng Xu. Improving global motion estimation in sparse IMU-based motion capture with physics. *ACM Trans. Graph. (TOG)*, 44(4):1–16, 2025.
- [154] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Vladislav Golyanik, Shaohua Pan, Christian Theobalt, and Feng Xu. Egolocate: Real-time motion capture, localization, and mapping with sparse body-mounted sensors. *ACM Trans. Graph. (TOG)*, 42(4), jul 2023.

- [155] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (PIP): Physics-aware real-time human motion tracking from sparse inertial sensors. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, 2022.
- [156] Xinyu Yi, Yuxiao Zhou, and Feng Xu. TransPose: Real-time 3D human translation and pose estimation with six inertial sensors. *ACM Trans. Graph. (TOG)*, 40(4), 2021.
- [157] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Physical non-inertial poser (PNP): Modeling non-inertial effects in sparse-inertial human motion capture. In *ACM SIGGRAPH Conf. Proc. (SIGGRAPH)*, 2024.
- [158] Handi Yin, Bonan Liu, Manuel Kaufmann, Jinhao He, Sammy Christen, Jie Song, and Pan Hui. Egohdm: A real-time egocentric-inertial human motion capture, localization, and dense mapping system. *ACM Trans. Graph. (TOG)*, 43(6), nov 2024.
- [159] Qijun Ying, Zehua Cao, Ziyu Wu, Wenwu Deng, Yuchen Zhong, Yukun Diao, and Xiaohui Cai. FoRM: Foot-driven reconstruction of human motion using dual-modal plantar pressure and inertial sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, 9(3), 2025.
- [160] H. Yonemoto, K. Murasaki, T. Osawa, K. Sudo, J. Shimamura, and Y. Taniguchi. Egocentric articulated pose tracking for action recognition. In *Proc. Int. Conf. Mach. Vis. Appl. (MVA)*, pages 98–101, 2015.
- [161] Y. Yuan and K. Kitani. 3D ego-pose estimation via imitation learning. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 763 – 778, Sep 2018.
- [162] Y. Yuan and K. Kitani. Ego-pose estimation and forecasting as real-time PD control. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 10081–10091, 2019.
- [163] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpo: Simulated character control for 3d human pose estimation.

- In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7155–7165, 2021.
- [164] He Zhang, Shenghao Ren, Haolei Yuan, Jianhui Zhao, Fan Li, Shuangpeng Sun, Zhenghao Liang, Tao Yu, Qiu Shen, and Xun Cao. MMVP: A multimodal mocap dataset with vision and pressure sensors. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 21842–21852, 2024.
- [165] Libo Zhang, Xinyu Yi, and Feng Xu. BaroPoser: Real-time human motion tracking from IMUs and barometers in everyday devices. In *ACM Symp. User Interface Softw. Technol. (UIST)*, 2025.
- [166] Peizhao Zhang, Kristin Siu, Jianjie Zhang, C. Karen Liu, and Jinxiang Chai. Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *ACM Trans. Graph. (TOG)*, 33(6), Nov. 2014.
- [167] Yufei Zhang, Jeffrey O. Kephart, Zijun Cui, and Qiang Ji. Physpt: Physics-aware pretrained transformer for estimating human dynamics from monocular videos. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2305–2317, 2024.
- [168] Yu Zhang, Songpengcheng Xia, Lei Chu, Jiarui Yang, Qi Wu, and Ling Pei. Dynamic inertial poser (DynaIP): Part-based motion dynamics learning for enhanced human pose estimation with sparse inertial sensors. In *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1889–1899, 2024.
- [169] Zhongyang Zhang, Kaidong Chai, Haowen Yu, Ramzi Majaj, Francesca Walsh, Edward Wang, Upal Mahbub, Hava Siegelmann, Donghyun Kim, and Tauhidur Rahman. Neuromorphic high-frequency 3D dancing pose estimation in dynamic environment. *Neurocomputing*, 547:126388, 2023.
- [170] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Hang Zhao, Tianhong Li, Antonio Torralba, and Dina Katabi. Through-wall human



- mesh recovery using radio signals. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 10112–10121, 2019.
- [171] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. RF-based 3D skeletons. In *Conf. ACM Spec. Interest Group Data Commun. (SIGCOMM)*, page 267–281, 2018.
- [172] Xiaozheng Zheng, Zhuo Su, Chao Wen, Zhou Xue, and Xiaojie Jin. Realistic full-body tracking from sparse observations via joint-level modeling. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023.
- [173] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *Int. J. Comput. Vis. (IJCV)*, 127:302–321, 2018.
- [174] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, pages 5738–5746, 2019.
- [175] Shihao Zou, Chuan Guo, Xinxin Zuo, Sen Wang, Pengyu Wang, Xiaojin Hu, Shoushun Chen, Minglun Gong, and Li Cheng. EventHPE: Event-based 3D human pose and shape estimation. In *IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 10976–10985, 2021.
- [176] Chengxu Zuo, Yiming Wang, Lishuang Zhan, Shihui Guo, Xinyu Yi, Feng Xu, and Yipeng Qin. Loose inertial poser: Motion capture with imu-attached loose-wear jacket. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2209–2219, June 2024.

# Achievements

## Journal

1. Ryosuke Hori, Ryo Hachiuma, Mariko Isogawa, Dan Mikami and Hideo Saito, Silhouette-Based 3D Human Pose Estimation Using a Single Wrist-Mounted 360° Camera, *IEEE Access*, vol. 10, pp. 54957-54968, May 23, 2022, doi: 10.1109/ACCESS.2022.3177623.
2. Ryosuke Hori, Mariko Isogawa, Dan Mikami and Hideo Saito, Event-PointMesh: Human Mesh Recovery Solely From Event Point Clouds, *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 31, no. 9, pp. 5593-5610, September 18, 2024, doi: 10.1109/TVCG.2024.3462816.
3. Ryosuke Hori, Hiroyuki Deguchi, Tsubasa Maruyama, Mitsunori Tada and Hideo Saito, Gait Inertial Poser (GIP): Gait-Aware Human Motion Capture Using Shoe-Embedded IMUs, *IEEE Access*, vol. 13, pp. 183262-183282, October 22, 2025, doi: 10.1109/ACCESS.2025.3624393.

## International Conference (Reviewed)

1. Ryosuke Hori, Ryo Hachiuma, Mariko Isogawa, Dan Mikami and Hideo Saito, Silhouette-Based Synthetic Data Generation For 3D Human Pose Estimation With A Single Wrist-Mounted 360° Camera,

*IEEE International Conference on Image Processing (ICIP)*, pp. 1304-1308, August 23, 2021, doi: 10.1109/ICIP42928.2021.9506043.

## Presentation

1. Ryosuke Hori, Privacy Preserving Human State Estimation, *1st International Workshop on Pseudo Privacy Preserving Data-based Human State Estimation, Measurement, and Applications (IW-P3HSE) Co-located with IEEE ISMAR 2024*, Invited Talk, October 21, 2024.
2. Ryosuke Hori, Mariko Isogawa, Dan Mikami and Hideo Saito, Event-PointMesh: Human Mesh Recovery Solely From Event Point Clouds, *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, TVCG Invited Presentation, March 11, 2025.
3. Ryosuke Hori, Ryo Hachiuma, Mariko Isogawa, Dan Mikami and Hideo Saito, Silhouette-Based Synthetic Data Generation For 3D Human Pose Estimation With A Single Wrist-Mounted 360° Camera (in Japanese), *Meeting of Image Recognition and Understanding (MIRU)*, Aug. 2021.
4. Ryosuke Hori, Mariko Isogawa, Dan Mikami and Hideo Saito, 3D Human Pose and Shape Estimation Using an Event Camera (in Japanese), *Information processing society of Japan Special Interest Group on Computer Vision and Image Media (IPSJ SIG-CVIM) Symposium*, Nov. 2022
5. Ryosuke Hori, Mariko Isogawa, Dan Mikami and Hideo Saito, 3D Human Pose and Shape Estimation Using an Event Camera (in Japanese), *Meeting of Image Recognition and Understanding (MIRU)*, Aug. 2023.
6. Ryosuke Hori, Mariko Isogawa, Dan Mikami and Hideo Saito, 3D Human Pose and Shape Estimation Using an Event Camera (in Japanese),

*Annual Conference of the Virtual Reality Society of Japan (VRSJ)*, Sep. 2023.

## Award

1. Ryosuke Hori, In *IPSJ SIG-CVIM Symposium*, SIG-CVIM Honorable Mention Award, Nov. 2022.
2. Ryosuke Hori, In *IPSJ SIG-CVIM Symposium*, Special Interest Group on Computer Graphics and Visual Informatics (CGVI) Student Presentation Award, Nov. 2022.
3. Ryosuke Hori, In *IPSJ SIG-CVIM Symposium*, IPSJ SIG-CGVI Excellent Research Presentation Award, Nov. 2022.
4. Ryosuke Hori, In *MIRU*, Student Encouragement Award, Aug. 2023.
5. Ryosuke Hori, In *VRSJ*, Academic Encouragement Award, Sep. 2023.