# Recovering 3D Human Motion in Scenes from Wearable Sensors

**Dissertation**

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von
Dipl.-Inform. Vladimir Guzov
aus Moskau/Russland

Tübingen
2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

| | |
|---|---|
| Tag der mündlichen Qualifikation: | 28.11.2025 |
| Dekan: | Prof. Dr. Thilo Stehle |
| 1. Berichterstatter: | Prof. Dr. Gerard Pons-Moll |
| 2. Berichterstatter: | Prof. Dr. Hendrik P. A. Lensch |
| 3. Berichterstatter: | Prof. Dr. Giovanni Maria Farinella |

To my family

# Abstract

In this thesis, we introduce novel methods for human motion and human-object interaction capturing from wearable devices using a capturing setup consisting of a head-mounted camera and body-mounted IMUs. While the vast majority of works in 3D human motion reconstruction have focused on capture methods from external cameras, we concentrate on wearable body sensors, which are often more scalable and easier to use. Through the course of our work, we solve the challenges of capturing motion from restricted and noisy data, modeling human-object interactions without visually observing them and reducing the capturing system to a single head-mounted device.

Before this thesis, wearable systems could recover human motion itself but could not localize it within large 3D scenes. As the first contribution, we introduce the Human POSEitioning System (HPS), the first system to enable long-term, high-accuracy 3D human pose estimation and self-localization within large scenes using the new wearable setup with a camera and IMUs. It combines deep learning-based camera localization with inertial pose estimation data and geometric clues from scene point clouds in the joint optimization algorithm. This enables HPS to recover the human motion and localize it within the scene while satisfying physical constraints such as foot-ground contacts. HPS demonstrates the feasibility of capturing extensive human motion data over extended periods, resulting in the collection of the HPS dataset – a dataset of long human activities in large scenes that established itself as a benchmark in the field. Together, the HPS system and dataset became a stepping stone for future research in wearable motion capture.

Our next contribution extends the capabilities of wearable systems further. We relax the primary assumption of HPS – static scenes – and present iReplica, a pioneering method to capture human-object interactions and model dynamic scene changes with a head camera and body-mounted IMUs. To solve the challenges of limited object visibility and human localization artifacts, we develop innovative algorithms for contact detection from motion and subject position correction from interactions. Together, these ideas allowed, for the first time, to model human-object interactions without external sensors. To train our model, we collected a new dataset comprising several hours of interaction data, contact timings, and first-person view camera videos, which we have

made publicly available to encourage further progress on the topic.

The aforementioned wearable system demonstrates great results but requires multiple body-mounted devices to capture human motion. Our next contribution reduces the number of sensors down to a single head-mounted device. While, at first, it seems impossible to recover the full body motion with such a minimalistic setup, our key idea is to use the information about the subject's surroundings to its fullest potential. We present HMD$^2$ – a first motion generation method that uses environment information obtained from the egocentric perspective in addition to the sparse motion input. It is powered by a diffusion-based motion model that generates human motion conditioned on video streams, device trajectories, and local scene point cloud reconstructions, all obtained from the same head-mounted device. Through our experiments, we demonstrate that the context of the scene and the device trajectory are sufficient to generate plausible human motion, closely matching the ground truth in most situations. HMD$^2$ greatly simplifies the hardware requirements and opens up new possibilities for applications in smart glasses and other minimalistic wearable technologies.

Overall, this thesis advances the field of wearable motion capture by addressing key challenges associated with egocentric capture systems and presents innovative solutions that blend multiple input modalities with novel fusion algorithms. The proposed methods and collected datasets pave the way for future research and practical applications in fields that require understanding and replication of human behavior, such as augmented reality, virtual presence software, and robotics.

# Kurzfassung

In dieser Dissertation stellen wir neuartige Methoden zur Erfassung von menschlichen Bewegungen und Mensch-Objekt-Interaktionen mithilfe von tragbaren Geräten vor, wobei wir ein Aufnahmesystem verwenden, das aus einer am Kopf montierten Kamera und am Körper montierten IMUs besteht. Während sich die überwiegende Mehrheit der Arbeiten zur 3D-Rekonstruktion menschlicher Bewegungen auf Aufnahmemethoden von externen Kameras konzentriert, liegt unser Schwerpunkt auf tragbaren Körpersensoren, die oft skalierbarer und einfacher zu verwenden sind. Im Laufe unserer Arbeit lösen wir die Herausforderungen der Erfassung von Bewegungen aus eingeschränkten und verrauschten Daten, der Modellierung von Mensch-Objekt-Interaktionen ohne visuelle Beobachtung und der Reduzierung des Aufnahmesystems auf ein einzelnes am Kopf montiertes Gerät.

Zuvor konnten tragbare Systeme nur die menschliche Bewegung erfassen, sie jedoch nicht innerhalb großer 3D-Szenen lokalisieren. Als ersten Beitrag stellen wir das Human POSEitioning System (HPS) vor, das erste System, das mithilfe des neuen tragbaren Setups mit einer Kamera und IMUs eine langfristige, hochpräzise 3D-Abschätzung der menschlichen Pose und Selbstlokalisierung innerhalb großer Szenen ermöglicht. Es kombiniert Deep-Learning-basierte Kameralokalisierung mit IMU basierter Positionsschätzung und geometrischen Anhaltspunkten aus Szenenpunktwolken in einem einzelnem Optimierungsalgorithmus. Dadurch kann HPS die menschliche Bewegung rekonstruieren und innerhalb der Szene lokalisieren, während physikalische Einschränkungen wie Fuß-Boden-Kontakte berücksichtigt werden. HPS demonstriert die Machbarkeit der Erfassung umfangreicher menschlicher Bewegungsdaten über längere Zeiträume, was zur Sammlung des HPS-Datensatzes führte – eines Datensatzes langer menschlicher Aktivitäten in großen Szenen, der sich als Benchmark in diesem Bereich etablierte. Zusammen wurden das HPS-System und der HPS-Datensatz zu einem Sprungbrett für zukünftige Forschungen zur tragbaren Bewegungserfassung.

Unser nächster Beitrag erweitert die Fähigkeiten tragbarer Systeme weiter. Wir lockern die Hauptannahme von HPS – statische Szenen – und präsentieren iReplica, eine wegweisende Methode zur Erfassung von Mensch-Objekt-Interaktionen und zur Model-

lierung dynamischer Szenenänderungen mit einer Kopfkamera und am Körper montierten IMUs. Um die Herausforderungen der eingeschränkten Objektsichtbarkeit und der menschlichen Lokalisierungsartefakte zu lösen, entwickeln wir innovative Algorithmen zur Kontakterkennung aus Bewegungsmustern und zur Korrektur der Position des Subjekts anhand der Interaktionen. Zusammen ermöglichten diese Ideen zum ersten Mal die Modellierung von Mensch-Objekt-Interaktionen ohne externe Sensoren. Um unser Modell zu trainieren, sammelten wir einen neuen Datensatz, der mehrere Stunden Interaktionsdaten, Kontaktzeiten und Videos aus der Egoperspektive umfasste, die wir öffentlich zugänglich gemacht haben, um weitere Fortschritte in diesem Bereich zu fördern.

Das oben erwähnte tragbare System zeigt großartige Ergebnisse, erfordert jedoch mehrere am Körper getragene Geräte, um menschliche Bewegungen zu erfassen. Unser nächster Beitrag reduziert die Anzahl der Sensoren auf einen einzigen am Kopf getragenen Sensor. Während es zunächst unmöglich erscheint, die gesamte Körperbewegung mit so einem minimalistischen Aufbau wiederherzustellen, besteht unsere Kernidee darin, die Informationen über die Umgebung des Subjekts optimal zu nutzen. Wir präsentieren HMD$^2$ – eine erste Methode zur Bewegungsgenerierung, die, zusätzlich zu den dünnbesetzten Bewegungseingaben, Umgebungsinformationen verwendet, die aus der egozentrischen Perspektive gewonnen wurden. Es basiert auf einem diffusionsbasierten Bewegungsmodell, das menschliche Bewegungen abhängig von Videostreams, Gerätetrajektorien und lokalen Szenenpunktwolkenrekonstruktionen generiert, die alle vom selben am Kopf getragenen Gerät stammen. Unsere Experimente zeigen, dass der Kontext der Szene und die Sensorbahn ausreichen, um plausible menschliche Bewegungen zu erzeugen, die in den meisten Situationen eng mit der Grundwahrheit übereinstimmen. HMD$^2$ reduziert die Hardwareanforderungen erheblich und eröffnet neue Möglichkeiten für Anwendungen in Smart Glasses und anderen minimalistischen tragbaren Technologien.

Insgesamt bringt diese Dissertation das Feld der tragbaren Bewegungserfassung voran, indem sie sich mit den wichtigsten Herausforderungen im Zusammenhang mit egozentrischen Erfassungssystemen befasst und innovative Lösungen präsentiert, die mehrere Eingabemodalitäten mit neuartigen Fusionsalgorithmen kombinieren. Die vorgeschlagenen Methoden und gesammelten Datensätze ebnen den Weg für zukünftige Forschung und praktische Anwendungen in Bereichen, die das Verständnis und die Nachbildung menschlichen Verhaltens erfordern, wie Augmented Reality, virtuelle Präsenzsoftware und Robotik.

# Acknowledgments

I want to thank everyone who has supported me during my Ph.D. journey.

I would like to express my deepest gratitude to my supervisor, Gerard Pons-Moll, for giving me the opportunity to work at the Max Planck Institute for Informatics and the University of Tübingen. His guidance and support have been invaluable throughout this journey. I extend my thanks to Bernt Schiele and Hendrik Lensch for their support as my second supervisors at the Max Planck Institute and the University of Tübingen.

A special thanks to Riccardo Marin, whose help I cannot overstate. As our postdoc, he provided insightful advice, actively contributed to projects, and offered unwavering support, especially during the demanding deadlines. His dedication and encouragement were essential in overcoming the toughest moments of my Ph.D. studies.

I am also grateful to Torsten Sattler for his valuable insights and assistance in our projects. His expertise greatly enriched my work. Additionally, I sincerely appreciate the support and collaboration of Lingni Ma, Yuting Ye, and the entire Surreal team, as I have learned so much from them.

I am fortunate to have collaborated with many brilliant minds, and I deeply appreciate the opportunity to work with Ilya Petrov, Aymen Mir, Yifeng Jiang, Xiaohan Zhang, Verica Lazova, Julian Chibane, Fangzhou Hong, Yannan He, Yunus Saracoglu, Richard Newcombe, C. Karen Liu and my other colleagues. Their insights and teamwork made the research all the more rewarding.

To my friends and colleagues at the Max Planck Institute for Informatics and the University of Tübingen, thank you for your support, discussions, and company. Your encouragement made this journey even more meaningful.

A big thanks to Connie Balzert and Jessica Endress, our assistants at MPI and University of Tübingen, for their help with all the administrative tasks.

Finally, my heartfelt thanks to my family and my girlfriend for their unwavering support throughout my studies. Their belief in me has been my greatest source of strength.

This work was made possible by funding from the Carl Zeiss Foundation. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans). This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Biological intelligence evolved to allow navigation and interaction with the environment [Gal90]. The human brain processes vast amounts of sensory information about the world, most of which is derived from our actions and movements. By moving and acting, humans gather information and achieve goals, making it essential to understand how we interact with the world to grasp human intelligence as a whole.

Each human action has a reason and thoughts preceding it. By studying human motion and learning to replicate it, we can infer the underlying intentions that led to actions. This would allow us to build models that are capable of simulating human behavior or better interpreting it.

If such models are developed, we can envision a future where people are able to interact with computers in a natural way, where robots can understand and replicate humans, where augmented reality systems provide assistance in everyday tasks, and where virtual meetings in the metaverse become a new way of communication. These advancements have the potential to revolutionize education, entertainment, manufacturing, and other aspects of our lives.

Bringing this vision to life requires capturing human motion and human-object interactions in a variety of real environments and situations, including cluttered apartments, large multi-room offices, outdoor spaces, and so on. This creates a need for a highly mobile system capable of motion and human-scene interactions capturing in diverse scenarios.

A standard way of capturing and understanding human motion has always been the use of external camera setups with multi-view pose estimation methods [HTTM12, NOT24]. While these methods offer high-quality captures, the environment where capture is possible is limited because every new scene would require a complicated process of setting up and calibrating multiple cameras. Besides that, the occlusion of body parts by an object or other body parts can result in a significant precision loss.

As an alternative to external capturing systems, wearable sensor setups have gained attention in recent years for several reasons: such systems are usually more affordable and have the advantage of increased mobility and faster setup time. Earlier works studied the ability to capture motion using several wearable setups, including inertial sensors [RLS09, PMBH+10] and inwards-looking cameras [RRC+16, TAP+20]. However, these solutions suffer from global position drift and the inability to localize the subject within the surroundings.

This dissertation proposes a new idea – capturing human motion and human-object interactions from a body-mounted sensor system consisting of an outward-looking camera and one or more inertial sensors. Such a system can localize the human in the scene and, as a consequence, reduce the motion drifting problem. In our work, we demonstrate that the system with this sensor layout can capture motion in large environments during an extended amount of time and model human-object interactions without external cameras or other external sensors. We have also found that, given the learned prior, it is possible to recover human motion even if the sensor layout is reduced to a single head-mounted device.

Of course, the task of egocentric capture comes with its own unique challenges: the two main sources of information – the video from the head-mounted camera and the inertial data – are mounted on the body and prone to distortions. Because of the head motion, the camera video is often blurry and does not provide a clear view of the scene. Moreover, the egocentric nature of the capture means that most of the subject's body and the object of interest (in case of interaction) are often out of sight. And the inertial data, as discussed earlier, is noisy and accumulates errors over time. Solutions to these challenges required us to combine several input modalities and develop novel data fusion algorithms.

In Chapter 4, we present *HPS* (Human POSEitioning System) – a system that enables, for the first time, reliable, long-term human motion capturing and body localization within the large 3D scene using only wearable sensors, namely head-mounted camera and body-mounted IMUs. We propose a novel optimization algorithm that combines deep learning-based camera localization with inertial pose estimation data and geometric clues from the scene point cloud. HPS is able to estimate the 3D pose of a human subject for longer periods of time and localize them in a large scene with high accuracy. HPS enabled us to capture the HPS dataset - a collection of more than three hours of various human activities captured in large laser-scanned 3D scenes. The direction set by HPS was explored more deeply in later works. For example, new methods study capturing motion using IMUs and body-mounted LiDAR [DLW+22] or sparser sensor setups

[JSM+23]. The wearable capturing setup used in HPS further proved its scalability and usefulness with large egocentric motion datasets such as Nymeria [MYH+24] – a collection of more than 200 hours of human motion and egocentric video from a head-mounted camera.

In Chapter 5, we further advance the capabilities of wearable setup with *iReplica* (Interaction Replica). iReplica addresses the primary limitation of HPS – its inability to capture dynamic human-object interactions. We present, for the first time, a method of modeling dynamic scene changes using only wearable sensors. The goal of extending wearable capture to dynamic objects brought several new challenges. The main challenge arises from the inability to model the interaction from the egocentric visual cues, as the object is frequently out of the camera's view. Therefore, our key idea in iReplica is a method for detecting human-object contact from the user's motion alone, without relying on unstable visual cues. However, contact detection is not the only insight that has enabled us to achieve the goal. We have also improved the HPS human localization pipeline to correct the trajectory based on the interactions with the objects. This makes interactions more realistic because it reduces the gap between the subject and the object. The two aforementioned algorithms, motion-based contact detection and object-based human localization correction, reinforce each other and serve as a fundament for human-object interaction modeling. As pioneers in the task, we collected a dataset of three hours of interactions, contact timings, and first-person view camera videos and made the data and the code online to foster the research in the field.

In Chapter 6, we introduce *HMD*$^2$ (Human Motion Diffusion from Head-Mounted Device), which simplifies the wearable capture system to a single head-mounted device with cameras and IMU, such as smart glasses. HMD$^2$ addresses the usability limitations of previous setups, eliminating the need to attach multiple IMUs to the body for each capture session. We show that precise and realistic motion can be recovered even from an extremely underconstrained single-device setup by leveraging the environment context and a motion model learned from the large amounts of egocentric motion data. At its base, our method relies on a diffusion-based motion generation model conditioned on a video stream from the camera, device trajectory, and a local scene point cloud reconstruction coming from the same device. The proposed method effectively bridges the gap between motion capture and motion generation: the system can follow the user when the information is sufficient to reconstruct the motion precisely, e.g., when the user's limbs are in sight, and generate plausible motion otherwise. Through extensive method evaluation on a large motion dataset, we show that HMD$^2$ can generate realistic motion for a variety of activities, such as walking, running, sitting, doing sports, and interacting with

objects.

This dissertation contributes to the field of human motion understanding by presenting novel methods and systems for motion capture from wearable devices. From the comprehensive multi-sensor HPS setup to the streamlined single-device solution, our work addresses key challenges and opens new pathways for egocentric motion capture research. By advancing the capabilities of wearable systems, we aim to bridge the gap between human and machine understanding of the world and accelerate the research in the field of egocentric vision. In the future, we hope to see the adoption of wearable devices as a standard solution for motion capture and analysis, as well as the development of broader applications such as AR interfaces, telepresence systems, and intelligent robotics.

## 1.1 Publications

The work presented in this dissertation has been published in the following papers:

- [GMSPM21] **Vladimir Guzov***, Aymen Mir* Torsten Sattler, Gerard Pons-Moll (* equal contribution) "Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-localization in Large Scenes from Body-Mounted Sensors", Conference on Computer Vision and Pattern Recognition (CVPR) 2021.

- [GCM+24] **Vladimir Guzov**, Julian Chibane, Riccardo Marin, Yannan He, Yunus Saracoglu, Torsten Sattler, Gerard Pons-Moll "Interaction Replica: Tracking human-object interaction and scene changes from human motion", International Conference on 3D Vision (3DV) 2024.

- [GJH+25] **Vladimir Guzov***, Yifeng Jiang*, Fangzhou Hong, Gerard Pons-Moll, Richard Newcombe, C. Karen Liu, Yuting Ye, Lingni Ma (* equal contribution) "HMD$^2$: Environment-aware Motion Generation from Single Egocentric Head-Mounted Device", International Conference on 3D Vision (3DV), 2025.

Contributions were also made to the following papers, which are not included in this dissertation:

- [ZBS+22] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, **Vladimir Guzov**, Gerard Pons-Moll "COUCH: Towards Controllable Human-Chair Interactions", European Conference on Computer Vision (ECCV), Springer, 2022.

- [LGO+23] Verica Lazova, **Vladimir Guzov**, Kyle Olszewski, Sergey Tulyakov, Gerard Pons-Moll "Control-NeRF: Editable Feature Volumes for Scene Rendering and Manipulation", Winter Conference on Applications of Computer Vision (WACV), 2023.

- [MYH+24] Lingni Ma, Yuting Ye, Fangzhou Hong, **Vladimir Guzov**, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David S. Fosas, C. Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, Richard Newcombe "Nymeria: A Massive Collection of Multimodal Egocentric Daily Motion in the Wild", European Conference on Computer Vision (ECCV), 2024.

- [ZBS+25] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Ilya Petrov, **Vladimir Guzov**, Helisa Dhamo, Eduardo Pérez Pellitero, Gerard Pons-Moll "FORCE: Dataset and Method for Intuitive Physics Guided Human-object Interaction", International Conference on 3D Vision (3DV), 2025.

- [HGK+24] Fangzhou Hong, **Vladimir Guzov**, Hyo Jin Kim, Yuting Ye, Richard Newcombe, Ziwei Liu, Lingni Ma "EgoLM: Multi-Modal Language Model of Egocentric Motions", ArXiv preprint 2409.18127, Sep 2024.

- [GPPM24] **Vladimir Guzov**, Ilya A. Petrov, Gerard Pons-Moll "Blendify – Python rendering framework for Blender", ArXiv preprint 2410.17858, Oct 2024.

- [ZSG+24] Xiaohan Zhang, Sebastian Starke, **Vladimir Guzov**, Zhensong Zhang, Eduardo Pérez Pellitero, Gerard Pons-Moll "SCENIC: Scene-aware Semantic Navigation with Instruction-guided Control", ArXiv preprint arXiv:2412.15664, Dec 2024.

# Chapter 2

# Background

## 2.1 Human body models

### 2.1.1 Kinematic tree skeleton animation models

Human body models are essential for various applications in computer graphics, computer vision, and robotics. These models provide a structured representation of the human body, enabling realistic animation, motion capture, and interaction with virtual environments. Typically, human models are represented by a skeleton and, optionally, a mesh driven by it. The skeleton is defined as a kinematic chain model. Kinematic chain models are used in computer animation and defined by a set of skeleton joints in a predefined standard position (*e.g.* T-pose) and their connections between each other. The connections follow a hierarchical tree-like structure, meaning every joint has one parent and one path to the central (root) joint. The transformations between the connected joints define the pose of the skeleton. In most cases, these transformations are limited to rotations. One can compute the transformation of each joint by following the kinematic chain from the root to the target joint and accumulating the transformations of each joint along the way.

To represent the human subject and their motion in our work, we use the Xsens skeleton model and the Skinned Multi-Person Linear (SMPL) body model [LMR+15]. We provide the description of these models below.

### 2.1.2 Xsens skeleton model

Xsens model of a body skeleton consists of 23 body joints, each with three degrees of freedom (DoF) of rotation and root joint having an additional three translation DoF. Within the capturing software, the skeleton respects the anatomical limitations of body

joint rotations. However, the skeleton model itself does not have any restrictions on joint rotation. The placement of the body joints is designed to be compatible with game character skinned skeletons commonly used in game engines, such as Mixamo [Mix24]. The body shape is encoded by the length of each bone in the kinematic tree and is calculated by the capturing software from the body measurements.

### 2.1.3 SMPL body model

SMPL is a differentiable function $M(\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{\beta}) : \mathbb{R}^{72\times3\times10} \mapsto \mathbb{R}^{6890\times3}$ that maps pose $\boldsymbol{\theta}$, translation $\boldsymbol{t}$ and shape $\boldsymbol{\beta}$ parameters to the vertices of a watertight human mesh. Compared to Xsens, which models only the underlying skeleton, SMPL features the complete body mesh. This is useful for modeling the collision of the body with the scene and creating a more realistic visualization.

The SMPL skeleton consists of 24 joints. The rotation of these joints is controlled by the vector of pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{72}$. Each triplet in the vector defines a relative rotation of the corresponding joint in axis-angle representation. The body mesh consists of 6890 vertices and is computed from the PCA coefficients of a shape space $\boldsymbol{\beta} \in \mathbb{R}^{10}$ inferred from a dataset of registered 3D scans. The pose of the mesh is controlled by the skeleton using skinning [MLT88] – an algorithm of transforming each vertex of the mesh according to the weighted sum of the transformations of skeleton joints.

In our work, we use $M_n(\boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{\beta}) \in \mathbb{R}^3$ to indicate the $n^{th}$ vertex of the SMPL mesh. We assume that the shape remains constant for each subject. Therefore, we set it once per subject using body measurements and drop shape parameter $\boldsymbol{\beta}$ for notational convenience.

## 2.2 Image data representations

General image representation involves creating embeddings or features from images that effectively describe their content. These embeddings are crucial for object detection, image retrieval, and scene understanding tasks. Selecting the right model for image representation depends on the specific requirements for spatial understanding, scalability, and domain. The resulting features can represent certain local regions of the image (local descriptors) and describe the image as a whole (global descriptors).

We use both global and local image descriptors in our work. Global image descriptors can effectively condense the image information into a compact representation, so we employ them in tasks of image filtering in the camera self-localization pipeline, described

in Chapter 4 and per-frame conditioning for the motion generation model in Chapter 6. Local image descriptors are employed in feature matching to achieve fine-grained camera positioning (Chapter 4). By describing the local regions of an image, they provide the necessary detail and robustness required for accurately matching features across different frames or viewpoints.

Below, we describe both types of descriptors in more detail.

### 2.2.1 Global image descriptors

Global image descriptors aim to encapsulate the overall content of an image into a compact, fixed-length vector representation. There are several techniques for forming the global descriptor of an image. Here, we will describe the ones used in our works. The first type is Vector of Locally Aggregated Descriptors (VLAD) descriptors, first formulated in [JDSP10], where the descriptor is formed by aggregation of features from the local descriptor points. Given local features descriptors of the image $F = \{f_1, \dots f_N\}$, $f_i \in \mathbb{R}^M$ we can form a bag of features $C = \{c_1, \dots c_K\}$ by clustering $F$ into $K$ clusters and computing the center of each cluster. Then, the VLAD descriptor is computed with the following formula:

$$\hat{v}^{iM+j} = \sum_{\{f : f \in F, \mathrm{NN}(f) = c_i\}} f^j - c_i^j, \tag{2.1}$$

where $f^j$ denotes $j^{\text{th}}$ element of $f$. After that, the resulting vector is normalized with $L_2$ norm $v = \frac{\hat{v}}{||\hat{v}||_2}$.

A different approach to forming a global image descriptor is to create the deep neural network model that generates the descriptors. The training procedure for this network varies based on the method and the data available. For example, the CLIP method [RKH+21] does so by maximizing the cosine similarity of the image descriptor model output and the text descriptor model output paired for the pairs of image and text descriptions. Another training procedure, shown in [CTM+21], is based on self-supervised learning from the batch of distorted views of one image. The idea is to have two models with identical architecture, so-called "student" and "teacher", with the objective of the student model to distill the probability distribution of the output of the teacher model. The teacher model is defined by the previous weights of the student model with an exponential moving average (EMA) applied to them.

### 2.2.2 Local image descriptors

Local image descriptors focus on representing specific regions or key points of an image, capturing detailed and localized information. They aim to preserve spatial details and provide representations that are invariant to transformations such as scale, rotation, and slight illumination changes.

The standard approach to forming local descriptors is to find a point of interest, or key points, and compute statistics on the region around this point, which will form the descriptor itself. Examples of commonly used statistics are a histogram of gradient orientations [DT05] or a binary vector of pixel intensities comparisons within the region [CLSF10].

The typical choice for a key point-finding algorithm is the one that identifies regions in an image where the intensity changes significantly in multiple directions. Notable examples include the Harris Corner Detector [HS+88], which uses the gradient information to detect corners by evaluating changes in intensity in a sliding window, and FAST (Features from Accelerated Segment Test) [RD06], which identifies key points by scanning a circle around a candidate point clockwise and examining pixel intensity patterns.

Recent advancements have introduced deep learning-based methods for generating local descriptors. For instance, the SuperPoint framework [DMR18] leverages a fully convolutional neural network to detect key points and compute their descriptors jointly. The model is trained in a self-supervised manner, ensuring robustness to diverse visual conditions. Such methods demonstrate superior performance in challenging scenarios, including extreme viewpoint changes or varying lighting conditions. However, the runtime of such methods is much slower than the statistics-based descriptors, which explains why the latter are still used in tasks sensitive to compute resource requirements, such as SLAM (Simultaneous Localization and Mapping) in AR and VR headsets.

# Chapter 3

# Related work

## 3.1 Human motion capturing

Capturing of the human motions can be done in various ways. The systems capable of capturing the body motion traditionally involved external cameras observing the actor. However, the recent progress in portable electronics has made it possible to build reliable and long-lasting wearable capturing systems. This section covers the related systems of human body capture, which uses external and wearable sensors.

### 3.1.1 External sensors setups

The predominant approach in vision has focused on analyzing humans using an external third-person camera, frequently disregarding the surrounding scene context [PMR11, KBJM18, OLPM+18, APMTM19, LGK20, SLAL20]. While some recent methods have begun capturing 3D scenes along with humans [HCTB19, SGXT20, LHG+23, LIYK22], they still rely on a third-person camera that observes a user from a distance. In addition, most of the methods do not support moving cameras, which limits the capturing volume.

By definition, all external camera methods suffer from occlusions – during the motion, body parts can become invisible to the camera because of the other people, objects, or the other body parts of the same subject. To overcome this problem, some methods use multiple cameras [SBB10, SC05, TWZ20, IBLM19, RHH+20], but this greatly increases the complexity of the setup because the cameras need to be synchronized and calibrated. In our work, we do not utilize external cameras to capture motion; instead, we use wearable systems. This allows us to capture the person in motion in various environments, including outdoors.

Despite the scalability drawbacks and complex setup, the multicamera external capturing methods can provide reliable human pose and shape data. For this reason, such

systems are used to evaluate our methods in Chapters 4 and 5.

### 3.1.2 Egocentric capture and motion generation

**IMU-based methods.** Inertial Measurement Units (IMUs) are becoming increasingly popular for capturing human motion due to their portability and low cost. Early work [VAV+07, RLS07] developed suits to capture 3D human pose during daily activities, consisting of IMUs placed on the key body parts. Despite the portability, these solutions require a large number of sensors to achieve accurate tracking, making them less practical.

Therefore, one line of work has focused on reducing the amount of IMUs necessary to capture motion. Such sparse IMU tracking methods demonstrated good performance while reducing the end-user burden. Using space-time optimization [vRBPM17] or data-driven approaches, these methods can track the full body using only six [YZX21, YZH+22] or four [YKL21, ZWZ+24] sensors.

Although commercial solutions for IMU-based pose estimation [RLS09, PSRB18] have improved the usability and stability of earlier solutions, IMU-based methods still suffer from drift, especially in the global orientation and location of the body. Therefore, they are not suitable for long-term tracking and capturing without additional sensors or external references.

**Camera-based methods.** A different approach is featured in many action recognition methods [BSAJ17, FFR11, MFK16, CZW+17, YMO+15, RSR15] and uses a system with a head-mounted camera which observes the user. The body pose estimation in this setup is mostly limited to the upper part. While solutions for full-body tracking exist [RRC+16, XCZ+19, TAP+20], the tracking precision is still low due to the high self-occlusion rate.

Some methods position the camera to face outward and estimate 3D pose from an egocentric view alone. However, these approaches often yield inaccurate results with high uncertainty [JG17, YK18, YK19]. As we demonstrate in Chapter 6, this problem can be tackled by introducing additional environment cues and visual-inertial camera tracking with IMU. As an interesting alternative, some methods mount multiple cameras on the body joints and apply structure-from-motion techniques [SPS+11], but these methods are currently restricted to capturing slow movements.

**Hybrid motion capturing methods.** To overcome the drift accumulation and uncertainty problems, several approaches combine IMUs with external cameras [vMPMR16, PMBH+10, PMBG+11, TGM+17, MVG+17], a depth-camera [HBB+13, ZYL+18] or

even a hand-held camera [vMHB+18, PMY+23]. However, they all require a camera that constantly observes the user, which limits the field of view to be captured or needs someone to follow the person being tracked.

In this dissertation, a wearable system with an outward-facing head camera and IMUs is considered (approximating the person's field of view), which is used to self-localize the person in the scene. In Chapters 4 and 5, we use a system with 17 IMUs and a head-mounted camera, which allows us to capture the person precisely. Later works demonstrated that it is possible to reduce the input point count to three [DKP+23, JSQ+22, CEJ+23] or even a single device [ZMZ+22]. Inspired by these results, in Chapter 6, we use a system with a single device with IMU and head-mounted cameras, improving the task of full body motion generation from head-mounted sensors.

### 3.1.3 Learned pose and motion priors

The evolution of capturing human motion has seen significant advancements through the incorporation of learned priors. Initial approaches focused on pose priors, such as VPoser [PCG+19] and Pose-NDF [TAL+22], which modeled human pose distributions to improve pose estimation accuracy and resolve ambiguities in captured data. These methods laid the groundwork for priors that expanded beyond static poses.

Subsequent research introduced motion priors, which capture temporal dynamics and constraints of human movement. For instance, methods like HuMoR [RBH+21] use learned motion representations to model realistic human trajectories over time, enhancing the consistency of captured sequences. These motion priors have proven particularly effective in applications where data sparsity or occlusions present significant challenges. Recent work incorporates diffusion models as priors, expanding capabilities in motion capturing from sparse inputs, as well as generation and editing. For instance, [STKB23a] enables long-sequence and interactive motion synthesis, while diffusion Noise Optimization (DNO) [KPA+24] serves as a universal prior for editing and control. In Chapter 6, we train a diffusion-based motion model to generate human motion from sparse data. During training, this model learns a human motion prior, helping to resolve ambiguities like the position of the body parts invisible to the camera.

## 3.2 Motion generation

### 3.2.1 Learning-based Pose and Motion Generation

Generating controllable and realistic human movements is a long-standing goal in computer graphics and vision. Modern deep learning opens new possibilities for this problem, with earlier attempts exploring both regression-based [HKS17, HKPP20] and generative [HAB20, LZCVDP20] frameworks. Recently, diffusion models demonstrated impressive capabilities in the generative setting across various tasks such as motion generation conditioned on text [STKB23b, ZDC+23], music [TCL23], audio [ANBH23] and sparse motion data [CEJ+23, DKP+23]. However, frameworks that generate motions in an online fashion with minimal latency [WLF+24] are still under-explored. Inspired by the use of autoregressive diffusion models for motion generation in other contexts [HPD+24, ZLAH23, SWJD23, YTY+23], in Chapter 6 we adopt diffusion-based motion generation with autoregressive inpainting for low-latency inference.

### 3.2.2 Scene-aware Pose and Motion Modeling

Motion generation and reconstruction satisfying scene and environment constraints is critical for learning-based motion models to become practical. Recent work has looked into various methods and representations to incorporate scene information, such as shape primitives [LIYK22, LJ23], point-cloud-based networks [HWL+23, ZWZ+22, WRL+22, WCL+22], voxel-based networks [WLX+23, HCV+21, SZKS19], scene images [CGM+20], signed distance fields [ZZW+23], to name a few.

However, most of the aforementioned methods cannot be scaled to large scenes and require end-of-motion goal specifications to navigate in the scene. To address these limitations, in Chapter 6, we represent the surroundings as a local patch of scene points encoded in a fixed-length vector using a pre-trained autoencoder.

## 3.3 Human-object interaction capturing

The majority of current methods that record human motion and human-scene interaction use external cameras. Methods to capture the humans in the environment assume mostly static scenes [HCTB19, ZZB+21, LIYK22, HYH+22] or work with a single dynamic object without any scene context [ZPJ+20, XJMS21, WY21, TGBT20, HTBT22, BXP+22]; similarly for human–scene and human–object interaction generation methods [TCBT22, ZBS+22, HCV+21, SZKS19]. A few exceptions exist, though, *e.g.*

RigidFusion [WLNM21] can track objects with an external RGBD sensor, and some pose estimation methods work with first-person view footage. However, those methods study the upper body or are limited to static cameras, *e.g.*, hand-object pose estimation [TBP19, KTS+21, LJX+21, DNMC20, OWL19, WPY+19], or do not model dynamic objects [ZYM+22]. Some methods can use egocentric video to predict the contact [DSZ+22, SGSF20] but do not further process this information to infer object position. In our case, the problem of the dynamic camera is complicated by the fact that the interacted object is often out of the camera's field of view. This makes it impossible to determine the start and the end of interaction from visual information. Therefore, pose-based interaction priors are more related to our case – one of the examples is Object pop-up [PMCPM23] – a method of predicting the object position based on the human pose alone. In Chapter 5, we follow a similar concept and infer the object position based on the contacts predicted from the human motion without visual cues.

## 3.4 Visual localization

Visual localization aims to estimate the pose of a camera in a known environment. Current state-of-the-art approaches are based on 2D-3D matches between pixels in the camera image and 3D scene points. These 2D–3D matches are either estimated based on local features [SCSD19, SDMR20, SLK17, SPGS18, IZFB09, LSHF12, LZA+20] or by regressing a 3D point coordinate for each pixel [SGZ+13, BR20, BR18, CGL+19, DFW+21]. A recent line of works focuses on the robustness of localization algorithms [TMT+20, WSG+20, JAG+21, YLZ21, DFW+21], *e.g.*, to illumination, weather, and seasonal changes, as well as to changes caused by human actions (rearranging furniture, *etc.*). These approaches assume that a large enough part of the scene remains static and is observed by the camera to facilitate pose estimation. The second assumption is violated in our scenario; therefore, the noisyness of the camera pose estimates is higher than in the standard localization scenarios. In Chapter 4, we address this issue by using spatial prefiltering of the camera poses before passing them in the optimization (Sec. 4.2.5).

It is also possible to use IMU-based tracking to bridge gaps where visual localization algorithms will most likely fail. This algorithm group is called visual-inertial odometry methods [LSB+15, JS11]. The data from the IMU sensor is especially helpful in stabilizing the predicted camera trajectory during periods of low scene visibility or when many scene changes are happening. Inspired by these methods, in Chapter 5, we use IMU data to stabilize the camera trajectory further, and in Chapter 6, we rely on the visual-inertial odometry system integrated into the capturing device to provide a precise

camera position.

Visual and visual-inertial localization methods are often used as a part of SLAM (Simultaneous Localization And Mapping) systems [MAMT15, CEG+21, WZ18, RACC20, Pro23]. SLAM systems are capable of building a map of the environment while simultaneously localizing the camera within this map. The map consists of the 3D points in the scene inferred by triangulating the 2D-3D matches from the camera images. This map can be used to compute the camera pose estimates, as well as to provide additional information about the scene. In Chapter 6, we use the scene points from the SLAM system to provide the scene context for the motion generation model.

# Chapter 4

# Human POSEitioning System (HPS)

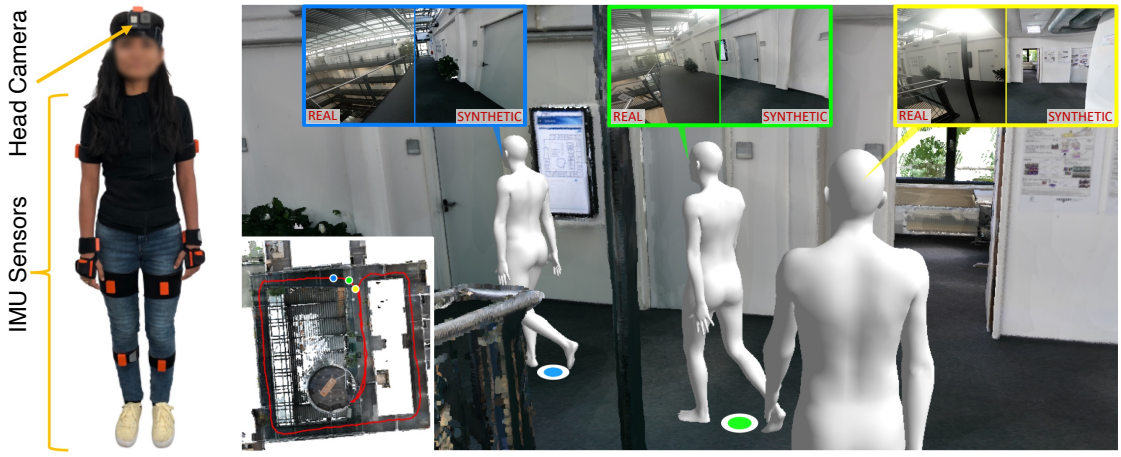



Figure 4.1: HPS jointly estimates the full 3D human pose and location of a subject within large 3D scenes, using only wearable sensors. Left: subject wearing IMUs and a head mounted camera. Right: using the camera, HPS localizes the human in a pre-built map of the scene (bottom left). The top row shows the split images of the real and estimated virtual camera.

In this chapter, we introduce the Human POSEitioning System (HPS), a method to recover the full 3D pose of a human registered with a 3D scan of the surrounding environment using wearable sensors. Using IMUs attached to the body limbs and a head-mounted camera looking outwards, HPS fuses camera-based self-localization with IMU-based human body tracking. The former provides drift-free but noisy position and orientation estimates, while the latter is accurate in the short term but subject to positional error accumulation over extended periods of time.

We show that our optimization-based integration exploits the benefits of the two, resulting in pose accuracy free of drift. Furthermore, we integrate 3D scene constraints

into our optimization, such as foot contact with the ground, resulting in physically plausible motion. HPS complements more common third-person-based 3D pose estimation methods. It allows for capturing larger recording volumes and longer periods of motion. Our system can be used for VR/AR applications where humans interact with the scene without requiring a direct line of sight with an external camera or to train agents that navigate and interact with the environment based on first-person visual input, like real humans. With HPS, we recorded a dataset of humans interacting with large 3D scenes (300-1000 $m^2$) consisting of 7 subjects and more than 3 hours of diverse motion.

This chapter is based on [GMSPM21][1], developed together with Aymen Mir with equal contribution. Aymen Mir was responsible for developing the optimization framework and IMU initialization, motion retargeting and alignment algorithms, while the author of this thesis was responsible for developing the image data capturing and processing pipelines, camera localization algorithm, method evaluation with external depth cameras, and rendering algorithms for visualization. The dataset capturing and processing responsibilities were distributed equally among both authors.

## 4.1 Introduction

Capturing the full 3D pose of a human, while localizing and registering it with a 3D reconstruction of the environment, using *only wearable sensors*, opens the door to many applications and new research directions. For example, it will allow Augmented / Mixed / Virtual Reality users to move freely and interact with virtual objects in the scene, without the need for external cameras. From the captured data, we could train digital humans that plan and move like real humans, based on visual data arriving at their eyes. Moreover, by relying only on egocentric data, we could capture a wider variety of human motion, outside of a restricted recording volume imposed by external cameras.

The prevailing approach in vision has been to analyze humans from an *external third-person camera* [PMR11, KBJM18, OLPM+18, APMTM19, LGK20, LHG+23, LIYK22]. Capturing with external cameras is undoubtedly a central problem in vision, but it has its limitations – occlusions are a problem, and interactions across multiple rooms or beyond the viewing area cannot be captured; consequently recordings are typically short.

We propose *Human POSEitioning System* (HPS), the first method to recover the full body 3D pose of a human registered with a large 3D scan of the surrounding environment relying *only on wearable sensors* – body-mounted IMUs and a head mounted camera, approximating the visual field of view of the human. Inspired by visual-inertial

[1] © 2021 IEEE. Reprinted, with permission, from [GMSPM21]

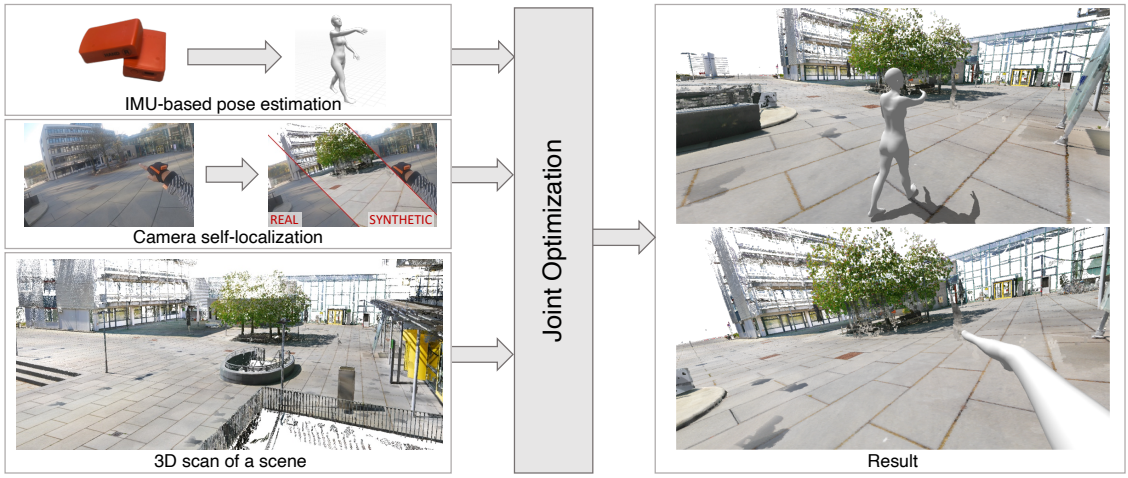



Figure 4.2: **Overview.** We use IMU data, RGB video from a head mounted camera, and a pre-scanned scene as input. We obtain an approximate 3D body pose using IMU data, and use head camera self-localization to localize the subject in the 3D scene. We then integrate the approximate body pose, the camera position and orientation, along with the 3D scene in a joint optimization to obtain the final location and pose estimates.

odometry and localization [LSB+15, JS11], as well as IMU-based human pose estimation [PMBH+10, vMHB+18, vRBPM17], HPS fuses information coming from body-mounted IMUs with camera pose obtained from camera self-localization [SLK17] [SCSD19, TOS+21] (see Fig. 4.1). Instead of placing the camera towards the body [RRC+16, TAP+20], we place it towards the scene, which allows us to capture what the human observes together with their 3D pose. In comparison to third-person pose methods, the body is not seen by the camera, which poses new challenges.

Pure IMU-based tracking is known to drift over time and camera localization produces many outliers. By jointly integrating IMU tracking with camera self-localization, we are able to remove drift [LSB+15, JS11], and recover the human trajectory when self-localization fails. Furthermore, since we can approximately locate the person in the 3D scene, we incorporate scene constraints when foot contact is detected. Overall, with HPS we recover natural human motions, registered with the 3D scene and free of drift, during *long periods* of time, and over *large areas*.

To demonstrate the capabilities of HPS, we capture a dataset of real people moving in large scenes. Our HPS dataset consists of 8 types of environments - some being larger than $1000m^2$, and 7 subjects performing a variety of activities such as walking, exercising, reading, eating, or simply working in the office. The dataset can be used as a testbed for egocentric tracking with scene constraints, to learn how humans interact and

move within large scenes over long periods of time, and to learn how humans process visual input arriving at their eyes.

We make the following contributions: **1**) to the best of our knowledge, HPS is the first approach to estimate the full 3D human pose while localizing the person within a pre-scanned large 3D scene using wearable sensors. **2**) we introduce a joint optimization which integrates camera localization, IMU-based tracking and scene constraints, resulting in smooth and accurate human motion estimates. **3**) we provide the *HPS dataset*, a new dataset consisting of 3D scans of large scenes (some larger than 1000 $m^2$), egocentric video, IMU data, and our 3D reconstructed humans moving and interacting with the scene. Unlike existing 3D pose datasets, typically captured from a third-person view, ours is captured from an egocentric perspective. We believe both HPS and HPS dataset provide a step towards developing future algorithms to understand and model 3D human motion and behavior within the 3D environment from an egocentric (or third-person) perspective.

## 4.2 Method

Our goal is to recover the 3D body pose and location of a subject in a known scene from egocentric measurements. To this end, our method requires as input: 1) a head-mounted camera, 2) body-mounted IMUs, and 3) a pre-built 3D scan of a scene, along with a database of RGB scene images with known camera parameters. Using camera data and a 3D scene reconstruction (Sec. 4.2.1), our method localizes the person within a pre-scanned 3D scene (Sec. 4.2.2), estimates their 3D pose using IMUs (Sec. 4.2.3), and in a joint optimization step (Sec. 4.2.4) integrates camera localization, IMU pose estimates and scene constraints, resulting in smooth and accurate human motion estimates. For an overview of our method, see Fig. 4.2.

### 4.2.1 3D Scene Reconstruction and RGB image Database

To obtain a representation of the 3D scene, we use a standard commercial solution to obtain a very dense scene point cloud: NavVis M6 [nav20] mobile capture system. It makes use of 4 LiDAR sensors and 6 RGB cameras. The scene is reconstructed from the LiDAR data and RGB images using a proprietary algorithm. The algorithm, in the process, also provides the extrinsic and intrinsic parameters for all the RGB images, which are later used for camera localization.

Potentially, other scene reconstruction methods using different sensors can be used

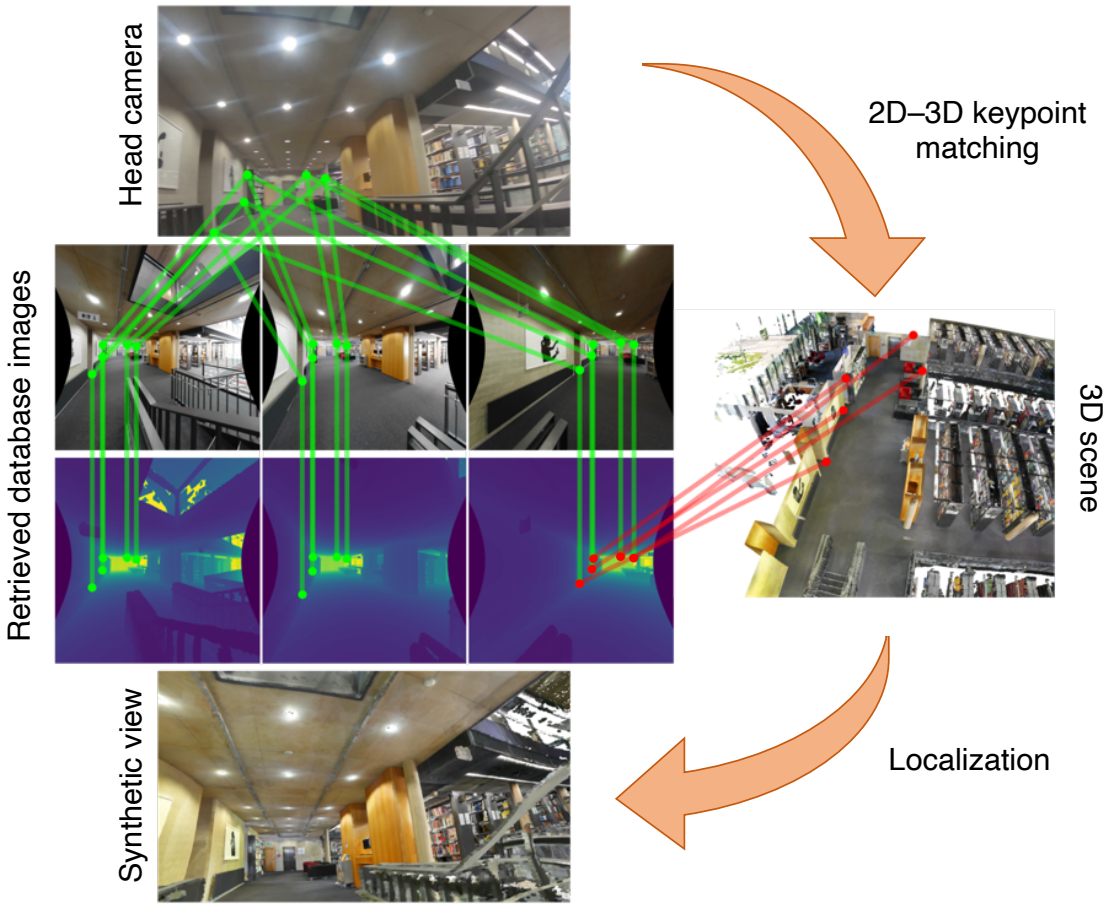



Figure 4.3: **Camera self-localization.** We match the head camera image keypoints with the keypoints from the prefiltered database with known 2D-3D scene correspondences. We then localize the camera in the scene by minimizing a reprojection error of the keypoints. *From top to bottom:* head camera image (query), top-3 retrieved images from a dataset, depthmaps rendered from the same position to map 2D database keypoints to 3D, synthetic view of the scene from the inferred camera position.

Figure 4.4: Comparison of a real image from a database and a result of our rendering

– the exact type of the 3D scanning method is unimportant as long as it registers the captured RGB images within the scene.

### 4.2.2 Camera Self-localization

The camera self-localization stage aims to estimate the position and orientation of the human head from a head-mounted camera. To scale to large scenes, we use a hierarchical structure-based localization algorithm [SCSD19, SDMR20] (Fig. 4.3). It first identifies a set of potentially relevant database images, *i.e.*, images used to build the 3D scene map, through image retrieval via global image descriptors. 2D-3D matches are established between local features extracted using a keypoint detector in the query image and 3D keypoints visible in the top-$K_{loc}$ retrieved images. To obtain 3D scene keypoints we produced a rendering of the scene pointcloud for each image in the dataset using known extrinsic and intrinsic camera parameters (Fig. 4.4). For rendering, we use the surface splatting technique [ZPVBG01] with a fixed splat size. Together with the color rendering, we produce a point mapping of the image. The resolution of the map is the same as the resolution of the color image. Each pixel of this map contains the index of the point in

the pointcloud, which is visible in that pixel. As our scanners use fisheye cameras, photos obtained from them are heavily distorted. This affects keypoint detection performance. To alleviate this, we undistort both camera images and point mappings by generating a mapping between our fisheye camera image plane and a fixed perspective camera plane. Undistorted images are run through the keypoint detector, and each keypoint is mapped to the position of the 3D point on the scan according to the aforementioned point mapping. 2D-2D matches between the query and the top-$K_{loc}$ retrieved database images thus yield the required 2D-3D matches.

These matches are then used to estimate the camera pose by applying a P3P solver [KSS11, HLON94, KBP10] inside a RANSAC loop [FB81] with local optimization [LMC12]. As a result, we obtain estimates for camera orientation $\mathbf{R}^C$ and position $\boldsymbol{t}^C$.

### 4.2.3 IMU based Pose Estimation

We use a commercial inertial motion capturing system from XSens [PSRB18], which uses 17 IMUs attached to the body with velcro straps or a suit. XSens IMUs provide 3D pose estimates, denoted as $\boldsymbol{\theta}^I$ and location estimates relative to the starting position of a recording - denoted as $\boldsymbol{t}^I$, using a proprietary algorithm based on a Kalman filter and a kinematic model of the human body to reduce drift. While it provides accurate articulation, our experiments show that the global orientation and position drift significantly over time, and consequently, scene constraints are not satisfied (Fig. 4.5, 4.9). Using acceleration information, IMUs also detect feet contacts with the ground, which we integrate in our joint optimization algorithm.

### 4.2.4 Joint Optimization

The HPS optimization algorithm aims to find the pose parameters of the parametric SMPL [LMR+15] body model, which satisfies the three conditions, namely:

- Camera localization results, which correlates to the head position;
- Scene constraints, imposed by the pre-scanned scene pointcloud;
- Motion temporal smoothness constraints, inspired by the physical properties of the motion, such as limitation on acceleration or the fact the there is normally no foot sliding while the foot is in contact with the surface.

Formally, the algorithm is a minimization task with the error function optimized with respect to the poses $\boldsymbol{\theta}_{1:T} \in \mathbb{R}^{72T}$ and the global translations $\boldsymbol{t}_{1:T} \in \mathbb{R}^{3T}$ of the body within

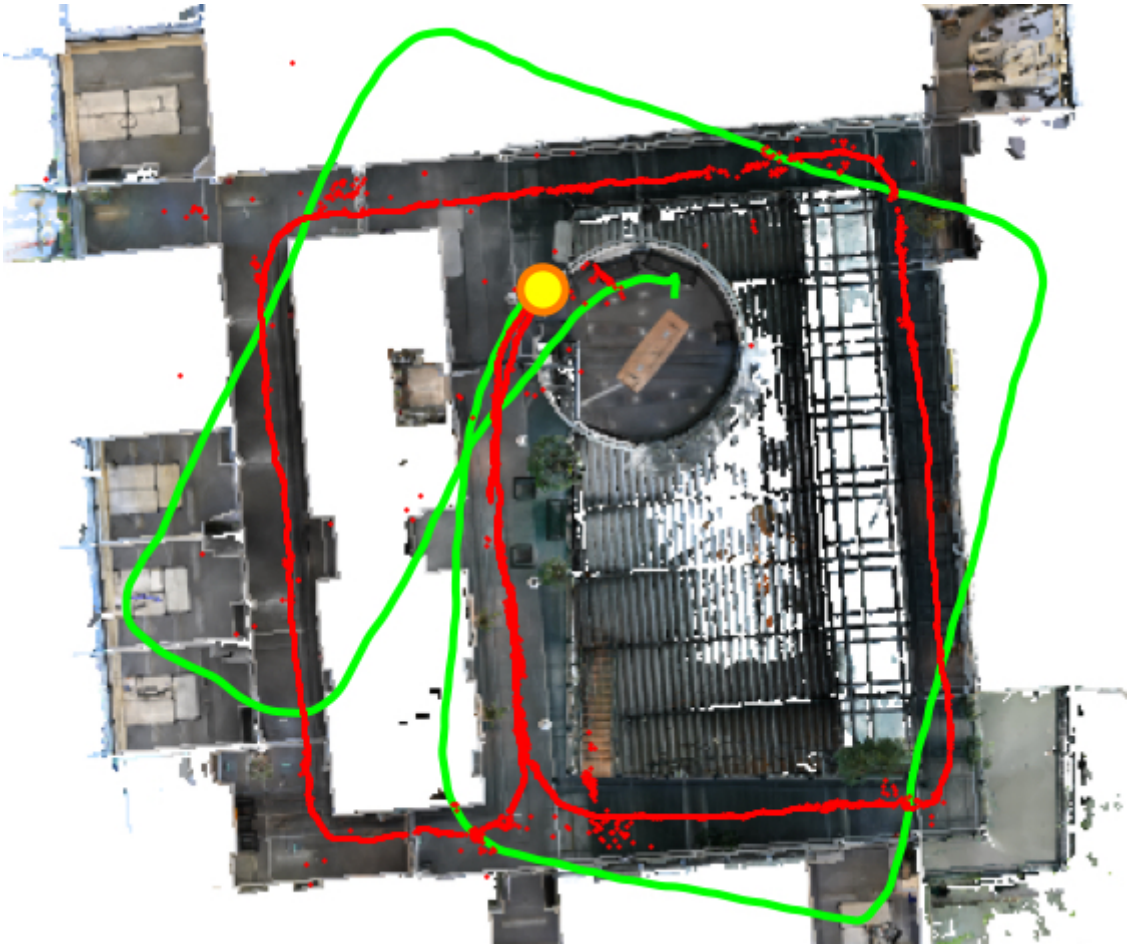


Figure 4.5: **Comparison of the trajectories** of IMUs (in green) with camera self-localization (in red). The yellow dot marks the start. Notice the red trajectory is free of drift but has outliers.

the window of $T$ frames long:

$$E(\boldsymbol{\theta}_{1:T}, \boldsymbol{t}_{1:T}) = w_{self} E_{\text{self}} + w_{scene} E_{\text{scene}} + w_{\text{sm}} E_{\text{sm}} + w_{IMU} E_{\text{IMU}}. \tag{4.1}$$

Below we provide the description of each loss term.

**Self-localization Term $E_{\text{self}}$:** The camera position, detected using the image localization method (Sec. 4.2.2), serves as noisy, but unbiased approximation of the head pose of the subject's body. This fact is used to constrain the motion in $E_{\text{self}}$ term, by minimizing the geodesic distance [vRBPM17] from the head camera orientation of SMPL model $\overline{R}^C(\boldsymbol{\theta})$, to the self-localization estimate $\mathbf{R}^C$ over $T$ frames:

$$E_{\text{self}} = \frac{1}{T} \sum_{j=1}^{T} ||(\log((\overline{R}^C(\boldsymbol{\theta}_j))^\top \mathbf{R}_j^C))^\vee||_2 \ , \tag{4.2}$$

where the log denotes matrix logarithm operation from Lie group $SO(3)$ to $so(3)$, and the $^\vee$ converts the skew-symmetric matrix to its axis-angle representation. Head pose $\overline{R}^C(\boldsymbol{\theta}_j)$ is obtained by first computing the head bone orientation $R^H : \mathbb{R}^{72} \mapsto \text{SO}(3)$ following the kinematic tree (see Sec. 2.1.1).

Next, a mapping to camera orientation is computed, using a constant camera to head offset estimated at frame 0, similar to previous works [vRBPM17, PMBH+10]:

$$\mathbf{R}_{HC} = (R^H(\boldsymbol{\theta}_0^I))^\top \mathbf{R}_0^C \ . \tag{4.3}$$

We find the desired mapping from pose to camera at a subsequent frame $j$ as

$$\overline{R}^C(\boldsymbol{\theta}_j) = R^H(\boldsymbol{\theta}_j)\, \mathbf{R}_{HC} \tag{4.4}$$

**Scene Contact Term $E_{\text{scene}}$:** A foot contact is detected by IMU capturing system as a peak in the acceleration of the leg [RLS09]. Using information about the feet contacts and the scene landscape, we can improve the realism of the motion by forcing the foot to stay in contact with the ground and prevent its sliding while the contact is detected. In HPS, we do this by using an error term consisting of two subterms:

$$E_{\text{scene}} = w_c E_{\text{contact}} + w_v E_{\text{slide}} \ . \tag{4.5}$$

We start by manually defining 4 vertex sets of the human model corresponding to the toe and heel regions of each foot: $\mathcal{B}_k$ with $k \in [1,2,3,4]$. Then, we can define $c_j^k \in [0,1]$ as a binary variable indicating if part $k$ is in contact with the ground at frame $j$.

We minimize the following contact error, which results in snapping the foot vertices to the closest scene vertices while the foot is in contact:

$$E_{\text{contact}} = \frac{1}{4T}\sum_{j=1}^{T}\sum_{k=1}^{4}\sum_{n\in\mathcal{B}_k}\frac{1}{|\mathcal{B}_k|}c_j^k||M_n(\boldsymbol{\theta}_j,\boldsymbol{t}_j)-v(n)||_2 \ , \tag{4.6}$$

where $M_n(\boldsymbol{\theta}_j,\boldsymbol{t}_j)$ denotes the $n^{th}$ vertex of the SMPL mesh at frame $j$ and $v(n) = \underset{\boldsymbol{v}_s\in\mathbf{V_s}}{\text{argmin}}(||M_n(\boldsymbol{\theta}_j,\boldsymbol{t}_j)-\boldsymbol{v}_s||_2)$ computes the closest scene point $\boldsymbol{v}_s \in \mathbf{V_s}$ to the $n^{th}$ vertex.

The motion of the foot in contact is also temporally constrained while in contact with the scene to prevent sliding:

$$E_{\text{slide}} = \frac{1}{4(T-1)}\sum_{j=1}^{T-1}\sum_{k=1}^{4}\sum_{n\in\mathcal{B}_k}\frac{1}{|\mathcal{B}_k|}c_j^k c_{j+1}^k||M_n(\boldsymbol{\theta}_j,\boldsymbol{t}_j)-M_n(\boldsymbol{\theta}_{j+1},\boldsymbol{t}_{j+1})||_2 \ . \tag{4.7}$$

**Smoothness Term $E_{\text{sm}}$:** Together with the contact term, we ensure the physical realism of the motion in one more way, by constraining the head pose, body rotation and global translation change between frames:

$$E_{\text{sm}} = w_T E_T + w_G E_G + w_H E_H, \tag{4.8}$$

where the translation term equals:

$$E_T = \frac{1}{T-1}\sum_{j=1}^{T-1}||(\boldsymbol{t}_j-\boldsymbol{t}_{j+1})||_2 \ . \tag{4.9}$$

and orientation terms are

$$E_G = \frac{1}{T-1}\sum_{j=1}^{T-1}||(\log((R^G(\boldsymbol{\theta}_j))^\top R^G(\boldsymbol{\theta}_{j+1})))^\vee||_2 \tag{4.10}$$

and

$$E_H = \frac{1}{T-1}\sum_{j=1}^{T-1}||(\log((R^H(\boldsymbol{\theta}_j))^\top R^H(\boldsymbol{\theta}_{j+1})))^\vee||_2 \tag{4.11}$$

with $R^G, R^H : \mathbb{R}^{72} \mapsto \text{SO}(3)$ as root and head rotation functions respectively, which are obtained from $\boldsymbol{\theta}_j$ pose parameters using the kinematic tree (Sec. 2.1.1).

**Pose Term $E_{\text{IMU}}$:** The pose recovered by IMUs $\boldsymbol{\theta}^I$ captures the articulation of the body

well, but is inaccurate for global orientation and translation, therefore we directly constrain the local pose parameters by the parameters derived from IMU capturing system.

$$E_{\text{IMU}} = \frac{1}{T}\sum_{j=1}^{T}\sqrt{(\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^I)^\top \mathbf{B}(\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^I)} \quad . \tag{4.12}$$

where $\mathbf{B}$ is an identity matrix with zeros at the diagonal entries corresponding to the root joint.

### 4.2.5 Initialization and coordinate frame alignment

Achieving convergence in the optimization of non-convex functions is highly dependent on the initialization. In our case, to solve the optimization problem stated in Eq. ((4.1)), we need to properly initialize the translation and the pose parameters of the body.

We start by initializing the translation parameters $\boldsymbol{t}_j$ using camera localization estimates $\boldsymbol{t}_j^C$. Given that camera localization results are often noisy, we detect and remove outliers by analyzing the translation velocity between each estimate and its inlier neighbors. A result is marked as an outlier if its velocity exceeds $3m/s$. This process is iterated until convergence, after which outliers are replaced via interpolation.

For pose initialization, the simplest approach is to use the IMU pose estimate, $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^I$. While this approach works reasonably well for the local joint angles, the global body orientation often diverges from the camera self-localization trajectory, which is more accurate on average despite the noise (see Fig. 4.5, 4.9). To address this, we align the initial tangent directions of the camera self-localization and IMU trajectories and let the optimization objective refine it further. The tangent directions are computed as

$$\boldsymbol{v}_j^C = \frac{\boldsymbol{t}_{j+\gamma}^C - \boldsymbol{t}_j^C}{||\boldsymbol{t}_{j+\gamma}^C - \boldsymbol{t}_j^C||_2} \quad , \quad \boldsymbol{v}_j^I = \frac{\boldsymbol{t}_{j+\gamma}^I - \boldsymbol{t}_j^I}{||\boldsymbol{t}_{j+\gamma}^I - \boldsymbol{t}_j^I||_2} \quad ,$$

where $\gamma = 10$ in our case. The root orientation $\widehat{\boldsymbol{\theta}_j^{I,G}}$ of the IMU pose is then corrected as follows:

$$\boldsymbol{\theta}_j^{I,G*} = (\log(\exp(\widehat{\boldsymbol{v}_j^I \times \boldsymbol{v}_j^C})\exp(\widehat{\boldsymbol{\theta}_j^{I,G}})))^\vee \quad , \tag{4.13}$$

where $\exp(\widehat{\boldsymbol{v}_j^I \times \boldsymbol{v}_j^C})$ is the planar rotation that aligns $\boldsymbol{v}_j^I$ with $\boldsymbol{v}_j^C$. For stationary frames, we use the correction derived from the last non-zero velocity frame.

Before the aforementioned initialization step, we need to align the IMU coordinate frame with the 3D scene reference frame. For that, we find a planar rotation $\mathbf{R}_A^*$ that

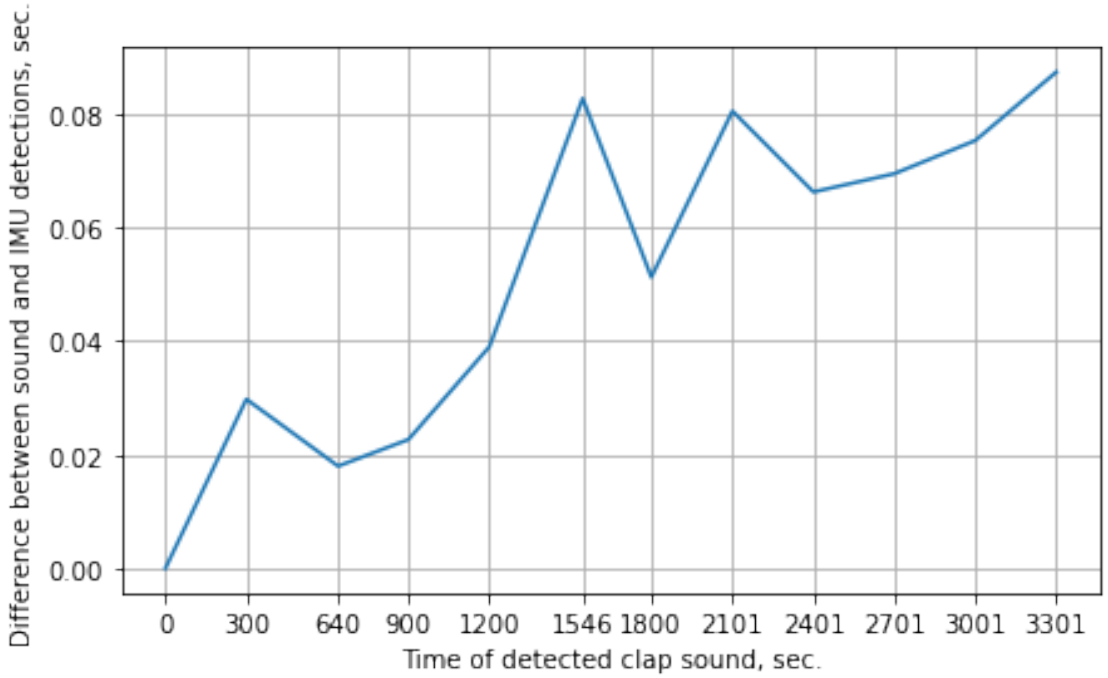



Figure 4.6: Measured time drift between camera and IMU clap detections after synchronization with the first clap.

aligns the SMPL head orientation at the very first frame $R^H(\boldsymbol{\theta}_0^I)$ with the corresponding camera orientation $\mathbf{R}_0^C$, by minimizing the following objective:

$$\mathbf{R}_A^* = \underset{\mathbf{R}_A \in \mathcal{R}}{\operatorname{argmin}} ||(\log(\mathbf{R}_A R^H(\boldsymbol{\theta}_0^I))^\top \mathbf{R}_0^C))^\vee||_2 \quad . \tag{4.14}$$

where the set of possible solutions is defined as $\mathcal{R} = \{\exp(\widehat{x\boldsymbol{\alpha}}) : x \in \mathbb{R}\}$ where $\boldsymbol{\alpha} = [0,0,1]^\top$ is the Z-axis (gravity) unit vector.

After alignment procedure, the IMU pose $\boldsymbol{\theta}_j^I$ and position $\boldsymbol{t}_j^I$ estimate of each subsequent frame are transformed to the 3D scene reference frame using the following formulas:

$$\boldsymbol{\theta}_j^{I,G} = (\log(\mathbf{R}_A^* \exp(\widehat{\boldsymbol{\theta}_j^{I,G}})))^\vee \quad , \boldsymbol{t}_j^I = \mathbf{R}_A^* \boldsymbol{t}_j^I \quad . \tag{4.15}$$

## 4.3 Implementation details

### 4.3.1 Joint optimization framework

Our optimization framework is implemented as a sliding window algorithm with a window size of 100 frames with 99 frames step. This means that windows do not intersect during the optimization phase, except for the last frame. This may introduce sudden jumps between the windows. To address this, we set the translation and orientation of the body on the first frame of the next window to be the same as the last frame of the previous window. The method is implemented in PyTorch using Adam optimizer [KB15] with 100 to 2000 optimization steps per window. Using Nvidia V100 GPU, one gradient step takes about 0.25 seconds.

### 4.3.2 Camera self-localization pipeline

For feature extraction and matching, we use SuperPoint [DMR18] and SuperGlue [SDMR20] PyTorch implementation pretrained on MegaDepth [LS18] and ScanNet [DCS+17] datasets. We detect 4096 keypoints at max for each image and perform 40 sinkhorn algorithm iterations at the matching stage. For database prefiltering we use NetVLAD PyTorch implementation pretrained on Pittsburgh 250k dataset [TSOP15]. We select $K_{loc} = 40$ most similar database images based on the cosine distance between query and database NetVLAD [AGT+16] descriptor. We use COLMAP [SF16, SZPF16] software to minimize the reprojection error of the matched keypoints. Overall, the pipeline takes around 3s to localize a frame at $1920 \times 1080$ resolution using Nvidia Q8000 GPU.

### 4.3.3 Camera calibration

To retrieve intrinsic parameters of the head-mounted camera, we take several photos of a checkerboard pattern and use OpenCV [Bra00] camera calibration tools to get the parameters. In our camera self-localization pipeline, we use an OpenCV camera model with 2 radial and 2 tangential distortion coefficients.

### 4.3.4 IMU-Camera synchronization

To synchronize IMU and camera data, we ask each subject to clap at the beginning of recordings. We detect these claps in both modalities to obtain synchronized starting

frames. To check for time drift between the two data streams, we performed the following experiment: we made a special 1-hour long recording with a subject clapping every 5 minutes. We synchronize our system with the first clap and measure the difference in time we get for each consecutive clap after that. We noticed a small accumulation of drift (Fig. 4.6) – around 87 ms per hour. We take this into consideration while performing long recordings.

## 4.4 Dataset

*HPS* allows us to collect the *HPS dataset* - a dataset of 3D humans interacting with large 3D scenes (300-1000 $m^2$, up to 2500 $m^2$). Our dataset contains images captured from a head-mounted camera coupled with the reference 3D pose and location of the person in a pre-scanned 3D scene. We capture 7 people in 8 large scenes performing activities such as exercising, reading, eating, lecturing, using a computer, making coffee, and dancing. All subjects have agreed to release their data for research purposes. In total, the dataset provides more than 300K synchronized RGB images coupled with the reference 3D pose and location. Figure 4.10 shows qualitative results from our dataset.

## 4.5 Experiments

This section shows that HPS does not drift with time and distance traveled, is robust to non-persistent camera localization outliers, and satisfies scene constraints (feet stay on the ground during contact and do not slide).

Since this is the first method to track humans in large scenes, there exist no published baselines to compare to, and ground truth 3D human pose and localization cannot be obtained for unbounded areas like ours. Hence, we use depth cameras to obtain ground truth dynamic point clouds of the human in a small sub-area of the scene. Subjects are then asked to move freely in the large scene and return to the sub-area, where we can evaluate accuracy and drift.

### 4.5.1 Quantitative Evaluation

**Ground-truth Point clouds.** We evaluate the accuracy of our method by comparing our output SMPL mesh (including translation) with a dynamic *ground-truth point cloud* of the person obtained from three synchronized and calibrated external Kinect depth

| Distance traveled | IMU | IMU + Cam | IMU + Cam (filtered) | HPS w\o scene | HPS |
|---|---|---|---|---|---|
| At start | 6.85 | 9.24 | 10.48 | 7.21 | **5.20** |
| 70 m | 54.49 | 742.32 | 6.93 | 6.48 | **4.60** |
| 200 m | 69.02 | 136.81 | 5.93 | 5.80 | **4.26** |
| 380 m | 108.44 | 32.17 | 6.15 | 5.69 | **4.53** |

Table 4.1: **Drift and camera outliers:** 3D error (in cm) for the subject standing in A-pose after moving freely around the scene.

| Distance traveled | IMU | IMU + Cam | IMU + Cam (filtered) | HPS w\o scene | HPS |
|---|---|---|---|---|---|
| At start | 6.77 | 2189.75 | 10.05 | 9.19 | **6.44** |
| 70 m | 51.57 | 569.71 | 21.75 | 20.68 | **15.96** |
| 200 m | 61.11 | 719.44 | 7.34 | 6.67 | **4.76** |
| 380 m | 100.44 | 261.72 | 12.59 | 11.96 | **10.07** |

Table 4.2: **Drift and camera outliers (dynamic):** 3D error (in cm) for the subject walking, standing and leaning on the table, after moving around the scene. Error is measured from the dynamic ground truth point cloud to the result (3D mesh in motion). Rows indicate distance traveled before evaluation.

cameras [Mic24]. We register the point cloud to the scene in three steps involving camera self-localization, ICP, and manual correction. The system is set up in a way that it covers all regions of the body and provides a 360° view of the subject every $1/30$ of a second (Fig. 4.7).

To obtain the RGBD videos from Kinects we use a custom recorder written in C++ that runs on a separate computer for each Kinect. To ensure time synchronization, Kinects are connected sequentially through the special time synchronization input and recorders are controlled over the wireless network. The video is recorded at 30 FPS with a color resolution of $1920 \times 1080$ pixels and a depth resolution of $640 \times 576$ pixels.

To register the ground truth point cloud to the static 3D prescanned scene, we use the following three-stage method:

- Visual localization: We obtain an approximate camera position for each depth camera using the same visual localization method as the one used for the head camera self-localization.
- Depthmap-to-scene ICP: We align the partial point cloud of each depth camera with the scene using ICP [BM92]. ICP is initialized using the camera parameters from the previous stage.

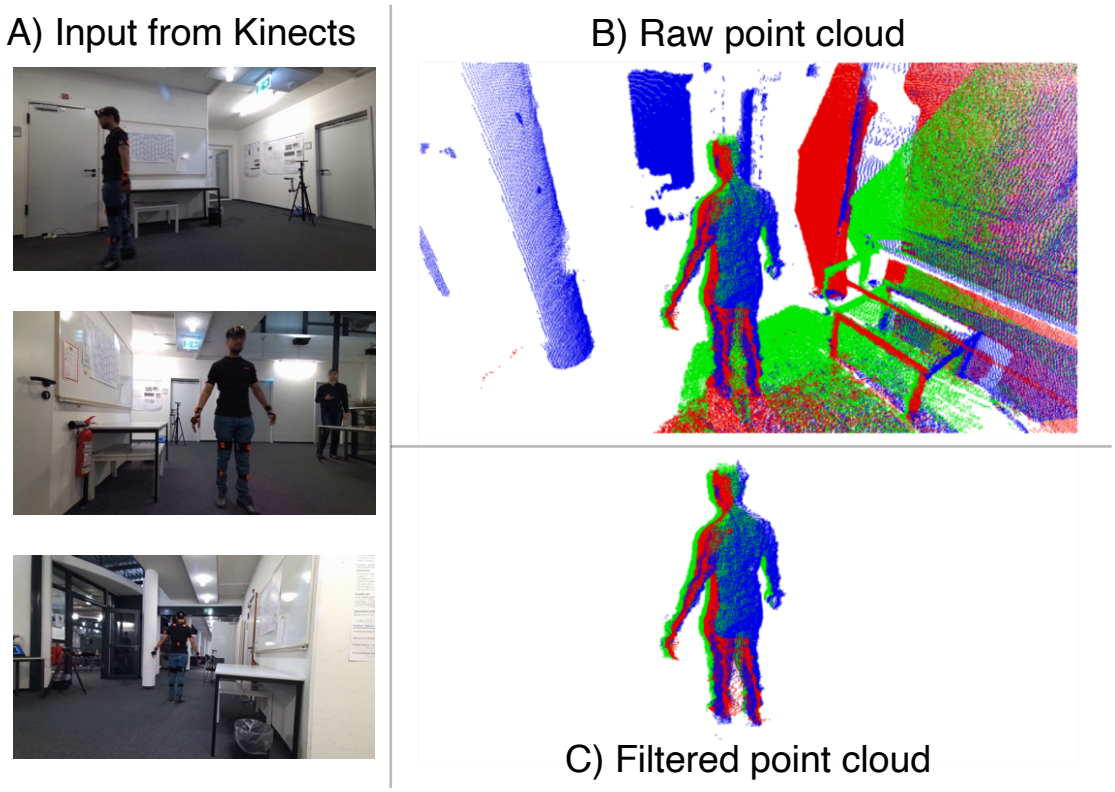



Figure 4.7: Setup of our ground truth recording system. A) Sample images captured by Kinects. B) The raw point cloud obtained by unprojecting the depth (RGB colors mark points from 1st, 2nd and 3rd Kinects respectively. C) The point cloud after applying background removal procedure.

| Metric | IMU | IMU + Cam | IMU + Cam (filtered) | HPS w\o scene | HPS |
|---|---|---|---|---|---|
| Distance to Surface | 188.38 | 39.8 | 0.95 | 0.32 | **0.056** |
| Foot Sliding | 0.92 | 52.09 | 1.75 | 2.00 | **0.90** |

Table 4.3: **Foot contact.** For frames when foot contact is detected, we report (in cm) **Distance to Surface** – average distance between foot vertices and the scene, and **Foot Sliding** – average distance on the surface plane between foot vertices in two successive frames. Numbers are computed for a 3 minute long walking sequence.

- Manual correction: Some results can still be erroneous due to depthmap artifacts or due to incorrect ICP initialization. These results are corrected manually. As a final step, we run ICP again.

**Video recording preparation.** Prior to the actual ground-truth video recording, we recorded a short video of the scene from all Kinects with no one present in the field of view. Frames are averaged, and average depth is used in 2 ways: in depthmap-to-scene ICP alignment and in background subtraction.

**Background subtraction.** To compute our metrics using obtained point clouds we need to make sure that we only get points corresponding to a subject. To achieve this, we first subtract the background from depth videos by ignoring all pixels with depth more or equal to the prerecorded empty scene depthmap. After that, we unproject the points from all 3 Kinects to the 3D space and run DBSCAN [EKSX96] clustering with the maximum distance between clusters of 0.12 meters. Finally, we remove all points that are not in the biggest cluster. An example of the result is shown in Fig. 4.7. We report the bidirectional Chamfer distance between the SMPL model (result) and ground truth point cloud from depth sensors *without* Procrustes alignment.

**Evaluation protocol.** For quantitative evaluation, we record using the following protocol: a subject starts within the recording volume of the three RGBD sensors and performs different actions including standing in A-pose, leaning on a table and walking. The subject then leaves the recording volume and moves within the scene, returns back and repeats the same actions inside that volume again. This is repeated several times, each time choosing a different path.

**Baselines.** There are no established baselines to compare to, as no other method tackles the same problem. Hence, to understand the influence of each component, we use the following baselines: 1) **IMU:** pure IMU tracker, 2) **IMU+Cam**: pose from IMU, and

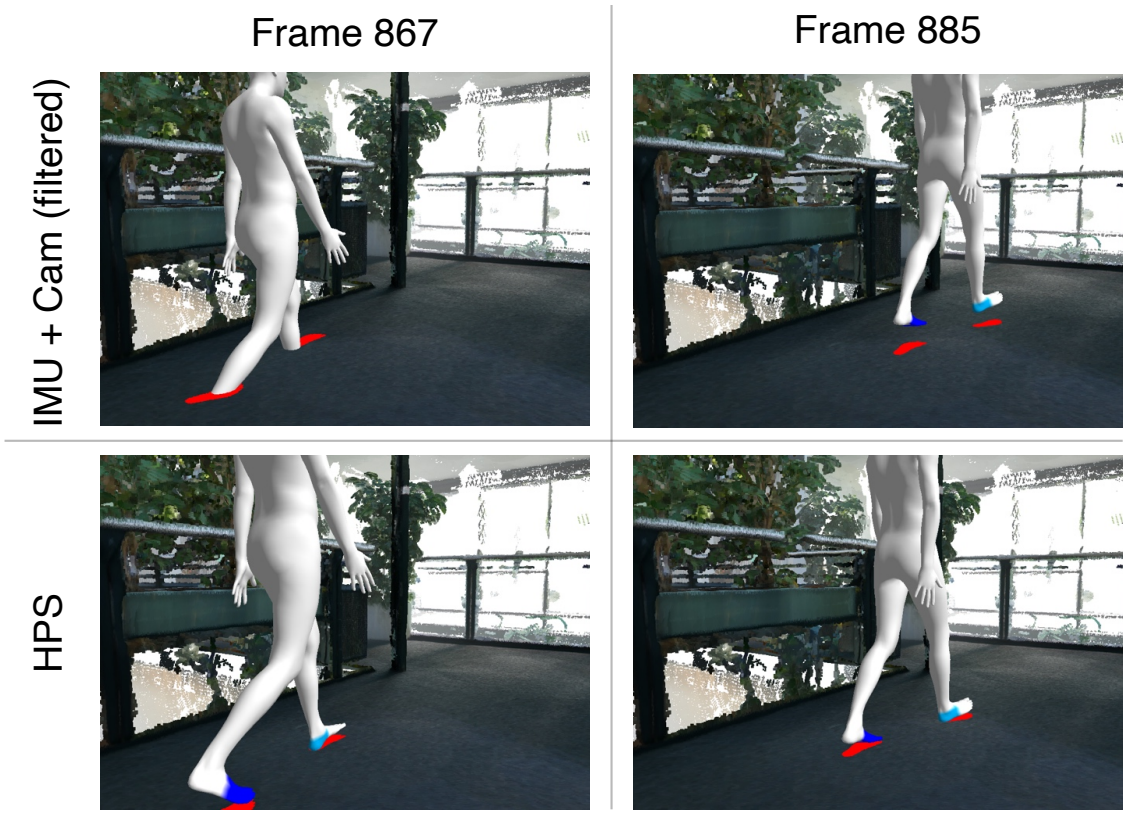



Figure 4.8: **Effect of integrating predicted 3D scene contacts.** As a baseline we used camera localization results for localizing SMPL model. Red regions mark closest surface to feet, heels and toes are colored with light blue and blue when IMUs detect ground contact.

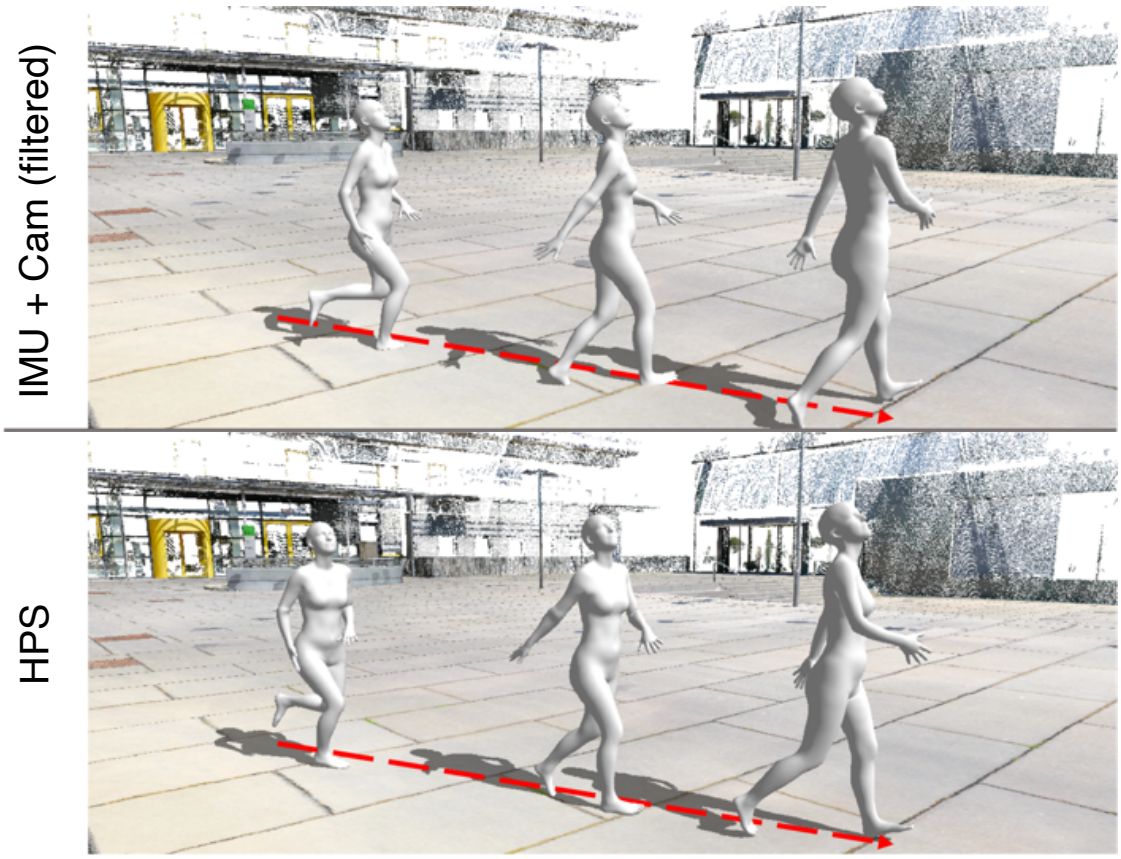



Figure 4.9: **Global body orientation improvement.** Combining the IMU pose with position from camera localization (IMU+Cam (filtered)) results in unnatural motion– the global body orientation does not face the direction of movement. By contrast, HPS correctly estimates the the global orientation.

translation from camera self-localization, 3) **IMU+Cam (filtered)**: Like IMU+Cam but with filtered camera outliers (same as in Sec. 4.2.5), 4) **HPS w\o scene**: Optimization without 3D scene contact constraints.

**Drift and Outliers.** In Tables 4.1 and 4.2, we compare HPS to the baselines. We observe that the IMU-only method drifts over time, particularly the global body translation and orientation. IMU+Cam corrects drift with camera localization, but produces translation noise and severe jitter. IMU+Cam (filtered) mitigates this, but lacks precision and suffers from global orientation errors (Fig. 4.9). HPS w\o scene further improves results, but without knowledge about foot-scene contacts, it is easily misled by incorrect camera localization, and the subject penetrates or flies over the ground. HPS results satisfy these scene constraints, and consistently achieve the best accuracy. HPS is inaccurate when

filtered camera localization fails for a long period (see 2nd and 4th rows of Table 4.2), but it can recover once the camera can be well localized in nearby frames (see 3rd row of Table 4.2). Overall, the analysis reveals that HPS does not drift (error does not increase with distance traveled or time), and is robust to non-persistent camera localization outliers.

**Foot contacts.** We also report in Table 4.3 the average foot-to-scene distance and foot-sliding-along-the-surface distance during contacts detected with the IMUs. HPS better preserves foot contact with the surface than the baselines, and has slightly lower foot sliding compared to the raw IMU tracker, which also integrates constraints with a *virtual* imaginary ground. Foot contacts in HPS result in stable and natural motion, see Fig. 4.8.

### 4.5.2 Qualitative evaluation

In Fig. 4.8 we show the effect of foot contact constraints. As we encourage contact with the scene surface each time a contact is detected, the human mesh does not fly in the air or penetrate the ground like the baseline. The motion is more stable and physically correct. In Fig. 4.10 we show examples of humans performing different actions including sitting, leaning on a table, dancing or performing push-ups.

## 4.6 Conclusions

We introduced HPS, to the best of our knowledge, the first method to estimate full body pose registered with a pre-scanned 3D environment from *only wearable sensors*. We demonstrate that HPS produces natural human motion, removes the typical drift of pure IMU based systems, and is robust to non-persistent camera localization outliers. HPS is able to continuously track humans in large scenes ($300-1000m^2$) including multiple rooms and outdoors.

The error of HPS does not accumulate with time or distance traveled. However, if camera localization is inaccurate for long periods of time, HPS performance deteriorates. This can be seen in the errors, which range from $4cm$ to $15cm$. Two factors influence localization accuracy: 1) Lack of features, 2) scene changes between the static 3D scan and the real images, captured from the head camera.

While HPS achieves a remarkable accuracy and stability, many applications will require errors in localization and pose of less than $1cm$. We envision many exciting research directions to improve HPS. First, a local map could be built on the fly to update

the large static scene with objects that move, and adding new objects. This would improve localization and allow interaction with dynamic objects.

It is not inconceivable that, in the future, a dynamic 3D reconstruction of the world will be stored on the cloud, and will be continuously updated from cameras worn by people [Pro24]. Second, camera localization could incorporate semantics [BCC+18, ZBLD19], e.g. detecting static and reliable objects. Third, while HPS integrates foot contacts, scene constraints with other body parts can further improve results. More powerful would be to learn a model to *anticipate human intent* to improve tracking. For example, we could detect when the person is about to sit on a chair, or about to grab an object. Conversely, HPS can be used to build models of environment interaction and navigation [MAO+19, XZH+18] from human captures consisting of several hours, as we believe natural behavior arises only during long recordings. Fourth, we want to combine HPS with virtual humans of appearance [PLPM20, BTTPM19, MAPM20, BSTPM20] to generate realistic data for training and evaluation of 3D human analysis methods.

HPS is the first step in a new exciting research direction. As the HPS dataset and code are released for research use, we hope it will foster new methods to perceive and model scenes and humans from an egocentric perspective.

Figure 4.10: **Qualitative results of our method.** Our method can localize and estimate the 3D pose of people performing activities as diverse as exercising, dancing, reading, sitting, eating, talking in a range of indoor and outdoor scenes, all *without* external cameras.

# Chapter 5

# Interaction Replica

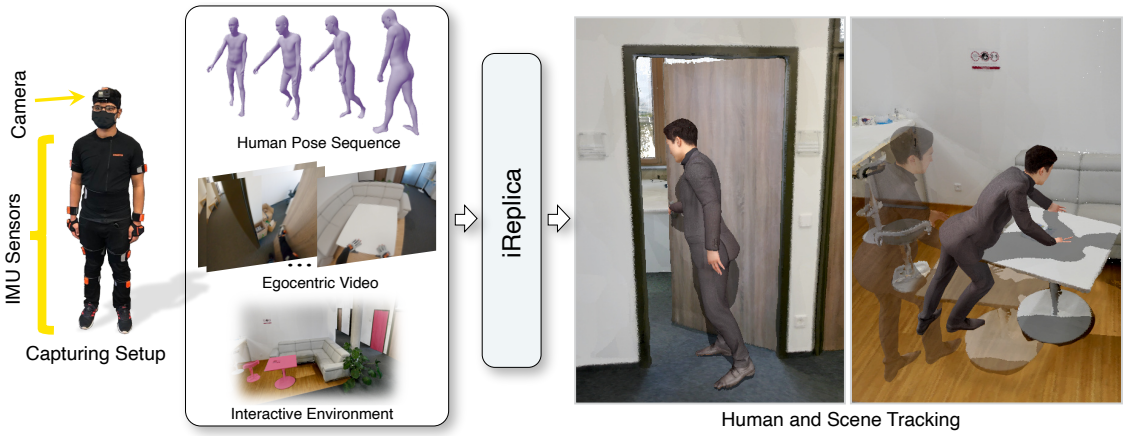



Figure 5.1: **Interaction Replica (iReplica)**. iReplica estimates location and full 3d pose of a subject within a large 3D scene and dynamically tracks changes made to the scene by the subject - using only wearable sensors (*left*), removing the need of external sensors. We obtain an approximate 3D human pose sequence using IMU sensors and use head camera self-localization to localize the subject in the prescanned 3d interactive environment scene. iReplica predicts human-scene contacts and updates the scene in case of interaction.

This chapter describes a follow-up of HPS, iReplica (published in [GCM+24][1]), which enables the wearable sensors system to capture human-object interaction for the first time.

Our world is not static and humans naturally cause changes in their environments through interactions, e.g., opening doors or moving furniture. Modeling changes caused by humans is essential for building digital twins, e.g., in the context of shared physical-virtual spaces (metaverses) and robotics. In order for widespread adoption of such emerging applications, the sensor setup used to capture the interactions needs to be in-

[1] © 2024 IEEE. Reprinted, with permission, from [GCM+24]

expensive and easy-to-use for non-expert users. This means that interactions should be captured and modeled by simple egocentric sensors such as a combination of cameras and IMU sensors, not relying on any external cameras or object trackers. To the best of our knowledge, no work tackling the challenging problem of modeling human-scene interactions via such an egocentric sensor setup existed before iReplica. This project closes the gap in the literature by developing a novel approach that combines visual localization of humans in the scene with contact-based reasoning about human-scene interactions from IMU data. Interestingly, we can show that even without visual observations of the interactions, human-scene contacts and interactions can be realistically predicted from human pose sequences. Our method, iReplica (Interaction Replica), is an essential first step towards the egocentric capture of human interactions and modeling of dynamic scenes, which is required for future AR/VR applications in immersive virtual universes and for training machines to behave like humans.

## 5.1 Introduction

Current augmented and virtual reality (AR/VR) applications show promising potential: interesting applications include collaborative developments, virtual meeting rooms, and personal assistants that help users navigate the world. While it is clear that for an immersive experience blending real and digital worlds is crucial, the current AR/VR experience is restricted to small spaces, i.e., in general, a few square meters, possibly free from objects. But consider daily actions like moving across rooms, opening and closing doors, or gathering chairs around a table. Even these simple actions are not easy to capture with present technology, which limits the scope of AR/VR applications.

The predominant approach for 3D human motion estimation relies on external cameras [KBJM18, JSS18, PZZD18, GNK18, IBLM19, KAB20, ZZB+21, LIYK22]. Yet, asking non-technical users to mount and calibrate complex multi-camera systems is clearly infeasible. Body-mounted sensors, *e.g.*, cameras and IMU sensors, seem much more ready for mass adoption.

Prior egocentric trackers such as HPS [GMSPM21], EgoLocate [YZH+23] or HSC4D [DLW+22] estimate human movement and position the person by combining head camera visual localization with IMU-based pose estimation. Methods like HPS, however, do not track scene changes. For example, if a person opens and walks through a door, such methods will only localize the person but can not infer the door movement, creating implausible reconstructions; see Figure 5.11, HPS.

In this work, we address, for the first time, the problem of human-scene interaction

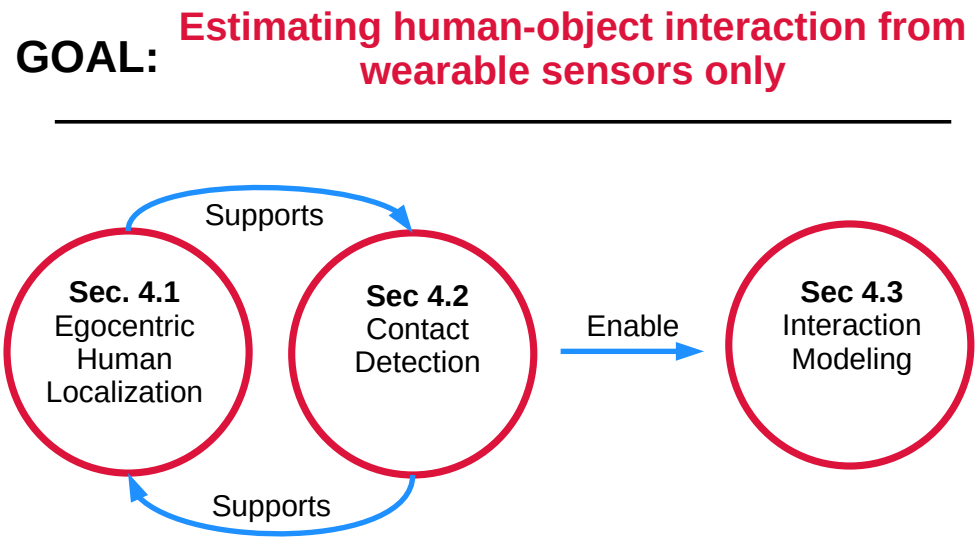



Figure 5.2: **Problem subdivision.** We demonstrate that joint integration of different sub-research problem improves and support each other. We show this is fundamental to achieve our goal of estimating human-scene interaction from wearable sensors only.

capture *from wearable sensors only*. Localizing the person with sufficient precision to track scene changes is hard, let alone estimating object motion. A major challenge is that the object is often not visible or is only partially visible in the camera; see Figure 5.3. In addition, since the head camera is in motion, the object's motion relative to the static world can not be directly inferred.

*Since no external sensors can measure the scene changes directly, how can we predict them?* Our key observations and findings are that 1) contact poses are distinctive and can be detected without visual clues, 2) knowing contact time stamps can regularize human localization, 3) objects move when the human contacts them[2]. Motivated by this, we propose *iReplica - Interaction Replica*, a novel human-centric method that automatically localizes the human in the scene (1. egocentric human visual localization), detects the time of contact and release with the object (2. contact time detection), and infers object motion based on contacts and human motion (3. interaction modeling). While works exist in each of these three sub-areas of research, no work integrates them simultaneously.

We needed several scientific innovations to integrate the aforementioned three sub-areas of research successfully. First, we improve human visual localization from Sec. 4.2.2 by optimizing the human trajectory to match reliable head camera poses and detect spatio-temporal contacts. Second, we train a transformer-based contact time de-

[2]In this work we only consider static objects moved by the captured human.

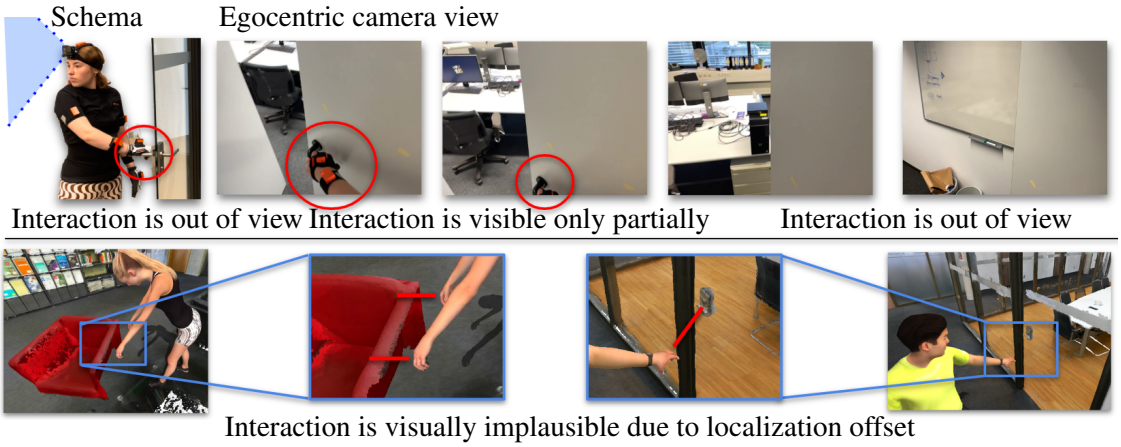



Figure 5.3: **Challenges.** Top row: The prediction of human-scene contacts (red circles) is hard because the interactions are frequently not in the camera view. Bottom row: Virtual replica of human pose and localization by prior work HPS [GMSPM21]. HPS achieved great progress in localizing humans solely by wearable sensors (camera+IMUs). However, for our task, the localization error of 4–16 cm (red lines) leads to visually implausible results for scene interactions.

tection approach based solely on the human pose, which achieves a remarkable accuracy of 0.91 and an average precision of 0.81. Third, based on the refined human visual localization in the scene and the accurate contact predictions, we infer object motion coherent with the human. Our results demonstrate that joint integration is beneficial (Fig. 5.2). The contact time information can be used to regularize visual localization by forcing the virtual human to contact the scene. Having precise human localization in the scene, along with contact timestamps, allows us to infer 1) where contacts occur and 2) the object's motion without seeing the object or contacts in the camera.

During this project, we captured two new datasets. To train a contact detection method, we captured a dataset of 8 subjects and more than 3 hours of human-scene interactions annotated with contact time stamps. To validate our proposed method, we captured a dataset with subjects moving and interacting with different objects in large 3D scenes. Our experiments show that iReplica can capture, for the first time, full interactions, including the human motion, its location within the 3D scene and the scene changes, all from wearable sensors alone. We demonstrate that our human-centric approach outperforms baselines, which rely on SOTA camera-based contact detection or visual object localization [DSZ+22, SGSF20].

In summary, our contributions are the following:

1. *Novel Problem & Method*: We are the first to tackle capturing human-scene interactions while localizing the human in the scene from wearable sensors alone. We

propose a method to address this problem, obtaining, *for the first time*, a digital replica of the human interaction in the scene without any external cameras.

2. *Novel Data & Metrics*: We provide H-contact – a dataset of 2300+ human-scene interactions with ground truth annotated contacts. Additionally, we provide Ego-HOI – a dataset of human-scene interactions in scanned environments. We propose metrics to measure the visual plausibility of reconstructed interactions and the accuracy of contact prediction and object localization.

To foster progress in this new research area, we release the method code, the evaluation protocol, and the datasets, including scans of the scenes, human motion capture aligned with them, and annotated contact timestamps.

## 5.2 Problem Setting

**Goal.** We aim to *estimate human-object interaction from wearable sensors only*, without information from external sensors, using only body-mounted IMUs and an egocentric camera. This opens a broad set of interconnected challenges: how do we define the interaction? How do we detect the start and the end of it? And how do we track the object's motion without having sensors dedicated?

**Assumptions.** We assume a static 3D scan of the scene, along with a set of marked interactive objects, knowing their initial position and degrees of freedom (*e.g.*, a sofa can slide on the ground but cannot be lifted, or a door rotates around a hinge). We refer to this as *interactive environment* (IE).

**Input/Output.** We require a set of body-mounted wearable IMUs (we use 17 sensors from XSens [PSRB18]) and a video stream from a head-mounted camera. Relying only on wearable sensors lets us handle large scenes consisting of multiple rooms. Compared to external cameras, wearable sensors are much more consumer-friendly as they are easier to set up. iReplica outputs a virtual replica of the interaction, *i.e.*, coherent human and object motion in the scene.

## 5.3 iReplica

**Overview.** We obtain initial localization and pose estimation for the person relying on an improved version of HPS [GMSPM21] (Sec 5.3.1, Fig 5.4 A). Our method considers only the human pose at each instant and predicts the probability of contact with an object

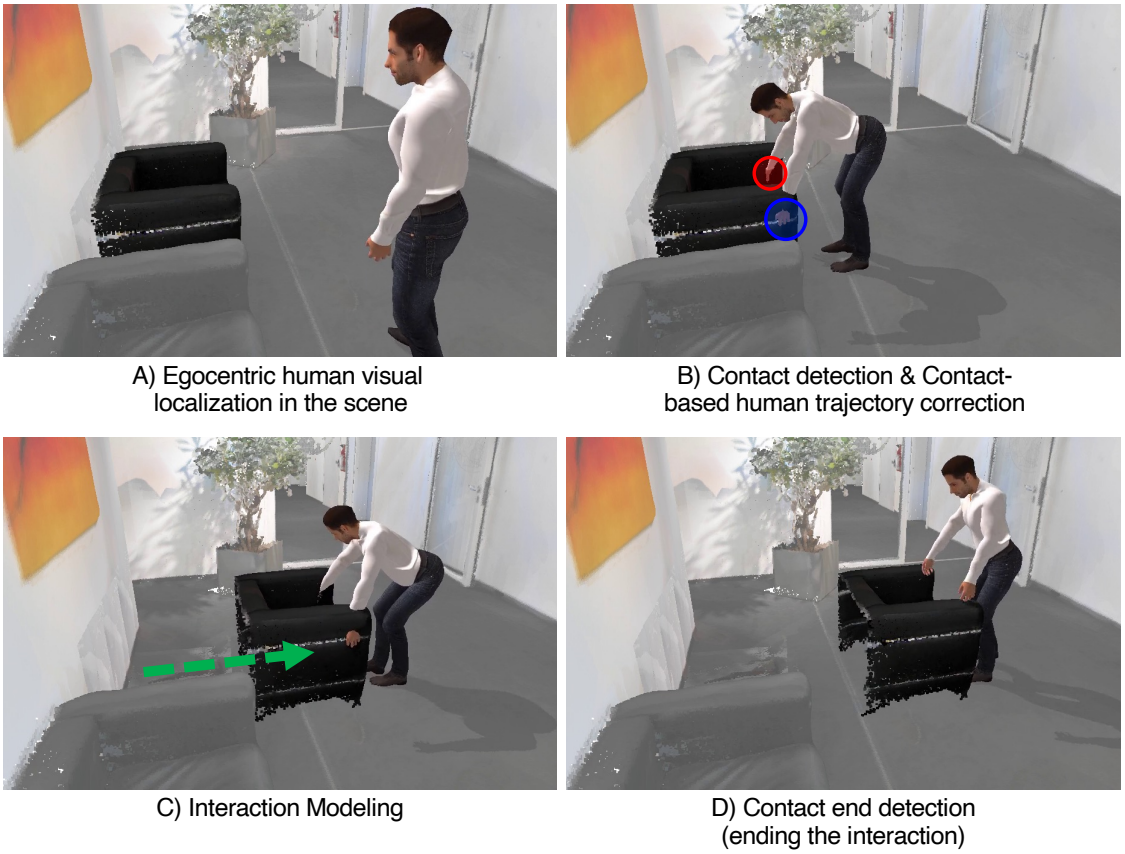



Figure 5.4: **Overview of iReplica.** iReplica estimates a subject's location and full pose within a large 3D scene and dynamically track changes made to the scene by the subject – using only wearable sensors. We do so in 4 steps: **A)** We obtain an initial localization of the subject in the IE by head camera self-localization. **B)** The start of the interaction is predicted by a neural network. Predictions are provided as contact / no-contact classification of the subject's hands (red and blue areas). The contacts are used to correct head camera localization of the subject, snapping the human trajectory smoothly to the object. **C)** The motion of the contacted regions is used to infer the object trajectory (green). **D)** The network predicts the release, essential to stop object dragging. The algorithm is detailed in Sec. 5.3.

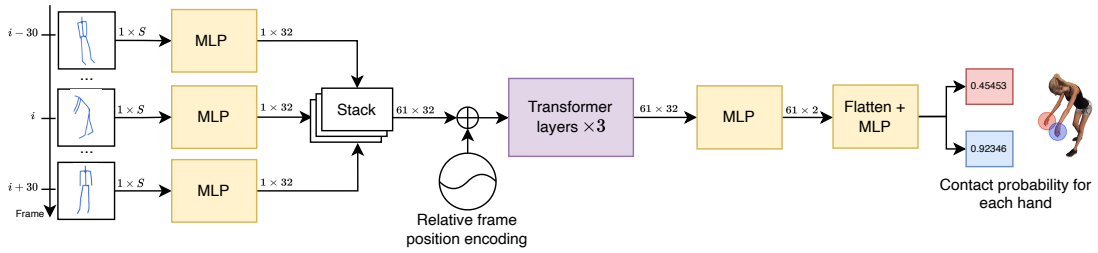



Figure 5.5: **Contact prediction based on human pose.** Interactions are frequently unobserved in an egocentric view, see Fig. 5.3 (top row), making contact prediction ill-posed. Instead, we propose to predict from sequences of full 3D human poses. We leverage a transformer-based architecture that takes 61 frames $\{i-30,\ldots,i+30\}$ of SMPL pose vectors of size $S=69$ and predicts the contact probability for each hand for the middle frame $i$. See Sec. 5.3.2 for details.

(Sec 5.3.2, Fig 5.4 B). Once the contact is detected, we model the object dynamically as follows: we deform the human trajectory to match the object contact; the object is attached to the human and driven in space according to its degrees of freedom (Sec 5.3.3, Fig 5.4 C); when our method infers from the human pose the end of the contact, the object is released (Fig 5.4 D).

### 5.3.1 Egocentric human visual localization.

**Problem.** Our method is built on a combination of IMUs and head-mounted camera data. Previous methods rely on optimizations to get from these two modalities smooth trajectories estimation [GMSPM21, JS11, LSB+15]. However, no previous approach considers the human's interaction with the scene nor shows extensions to incorporate constraints coming from this. Also, if 10 cm of error (average for HPS) might not seem much for human localization in a building, for human-object interaction (which is our ultimate goal), this can cause dramatic inconsistencies. Instead, we see (and take advantage of) the relation between human localization and contact prediction: solving for contact prediction supports human localization in large volumes; human localization helps detect object contact in time and space.

**Trajectory optimization.** We start introducing an improvement over the HPS approach. We deploy a simple optimization that is flexible and can be used to incorporate interaction constraints. While we work with 3D trajectories, we consider a 2D optimization since one dimension (gravity axis) is constrained by the ground of the scene. Consider the trajectory described as a 2D curve $\mathbf{l}(\tau) = (x(\tau), y(\tau))$ defined in the time interval

$\tau = [\tau_{\text{start}}, \tau_{\text{end}}]$, and a list of $K$ control points $\mathbf{p} = \{\mathbf{p}_i = (x_i, y_i)\}_{i=1}^{K}$ (constraints) at times $\tau_{\text{start}} \leq \tau_1, \ldots, \tau_K \leq \tau_{\text{end}}$. We want to recover a new trajectory $\hat{\mathbf{l}}(\tau) = (\hat{x}(\tau), \hat{y}(\tau))$ that gets close to the control points while not deviating too much from the initial trajectory. We introduce an energy $E_{tr}$ that encodes the trajectory deviation in terms of angles.

$$E_{tr}(\hat{\mathbf{l}}, \mathbf{l}) = \int_{\tau_{\text{start}}}^{\tau_{\text{end}}} \left( \frac{d\hat{\alpha}(\tau)}{d\tau} - \frac{d\alpha(\tau)}{d\tau} \right) d\tau, \tag{5.1}$$

where:

$$\hat{\alpha}(\tau) = \text{atan2}\left(\frac{d\hat{y}}{d\tau}, \frac{d\hat{x}}{d\tau}\right), \quad \alpha(\tau) = \text{atan2}\left(\frac{dy}{d\tau}, \frac{dx}{d\tau}\right).$$

Concretely, $E_{tr}$ measures the difference between two trajectories at each instant in terms of direction (angle) variation. We define the difference only in terms of angles since, as pointed out in HPS [GMSPM21], the total distance tracked by the IMUs is well measured, while the curvature tends to accumulate drift over time.

We then correct the human trajectory by optimizing the following energy:

$$F_{tr}(\mathbf{l}, \mathbf{p}) = \arg\min_{\hat{\alpha}} \left( \sum_{i=1}^{K} (||\hat{\mathbf{l}}(\tau_i) - \mathbf{p}_i||_2) + \lambda E_{tr}(\hat{\mathbf{l}}, \mathbf{l}) \right), \tag{5.2}$$

where $\lambda$ is the global rigidness coefficient, which encodes how much local angles should retain the initial estimation: a lower value lets the trajectory freely bent to fit the control points, while a higher value preserves the local curvature and promotes global rigidness of the trajectory. To help the reader's intuition, we report in Fig. 5.6 an example for different values of $\lambda$. atan2 function is defined as

$$\text{atan2}(y,x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases}$$

**Contact-based human trajectory correction.** In iReplica, we perform the above optimization two times. We consider the input trajectory recovered by the IMUs, and we optimize it using the control points returned by the camera localization. Then, our method detects the moments of contact along the human motion sequence. For each detection,

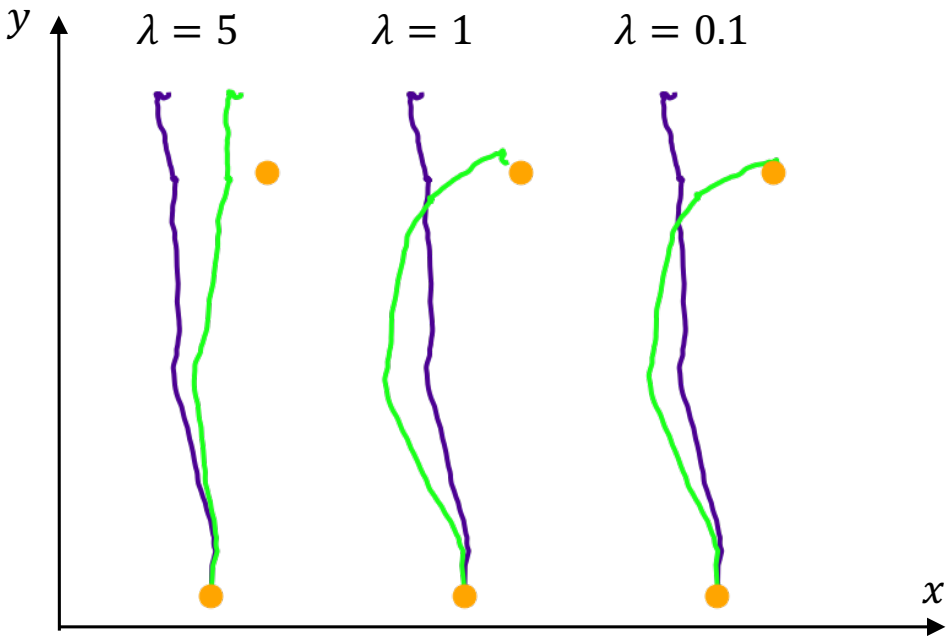



Figure 5.6: **Trajectory fitting with bending energy.** Bending trajectories with $F_{tr}$ using different rigidness coefficients $\lambda$, purple marks the original trajectory, green marks the result, orange dots denote control points.

we select the nearest object in the scene within a reasonable range (*e.g.*, 50 cm). The contact is ignored as a false positive if no object is that close. We select a contact point $\mathbf{p}_c$ as the closest point of the object to the contacting hand. Then, we rerun our optimization again, considering $\mathbf{p}_c$ as the only control point to satisfy.

### 5.3.2 Contact detection

**Problem.** The key ingredient for accurate localization is detecting when and where the user interacts with an object. In this work, we purely focus on human poses (obtained from IMUs) for contacts instead of relying on camera data for multiple reasons: (1) contacts are often not visible, *i.e.*, camera data alone is insufficient for the task. (2) IMUs are cheaper and much more power-efficient than cameras. At the same time, processing their lower-dimensional output requires significantly less compute (and thus power). This makes purely IMU-based contact prediction very attractive for applications running on mobile devices such as AR/VR headsets or robots. Naturally, combining inertial and visual data should improve performance, similar to visual-inertial localization. However, we leave this integration for future work and focus on IMU-only contact detection.

**Training data.** Existing datasets for human-object contact prediction contain only a limited number of samples per object type [BXP+22], or only hand-held objects [TGBT20]. In our context, the interaction involves large objects appearing in real scenes. Hence, we collect and annotate a training dataset (H-contact, Sec. 5.4.1) of ~680k pose frames (> 3 hours) recorded with 8 subjects wearing IMUs and 12 different objects. Our dataset is noticeably bigger compared to several other human–object and human–scene interaction datasets (BEHAVE [BXP+22] contains ~15k frames, PROX [HCTB19] ~100k).

**Transformer-based architecture.** To predict contacts, we train a sequence-to-sequence Transformer [VSP+17] to map a sequence of poses to a sequence of per-hand contact probabilities. Specifically, we concatenate 61 SMPL pose vectors of size $S = 69$ in a sequence, forwarding them to an MLP, appending the frame position as positional encoding, and processing them with a Transformer to output a sequence of contact probabilities for each hand. We use a sliding-window approach, and at each instance, we retain only the central ($30^{\text{th}}$) prediction. The contact is considered active once the probability reaches a certain threshold. The architecture is visualized in Fig. 5.5. To remove false negatives, any gap of $\leq 0.5$ s between two active contacts is filled (*i.e.* marked active). This produced the best results on a validation set – see experiments Sec. 5.4.4.

While focusing on hands is not entirely descriptive of how humans interact with the world, it covers most cases in which humans cause changes in their environments. Our method can easily extend to other body parts; more detailed analyses are left for future works.

**Contact intervals.** Each group of consecutive frames with active contact is considered a *contact interval*. If the network predicts the end for one hand while the other is still considered to be in contact, iReplica splits the contact interval into two interactions (a two-handed and a one-handed one). Similar cases (*e.g.*, interchanging hands) are treated the same way. Likewise, our method can handle multi-object interaction.

**Multiple Objects Interaction.** The contact strategy described above naturally extends to interaction with multiple objects. When the start of contact is identified, iReplica considers the closest object within a 0.5 m radius (if any) and initiates the interaction, which lasts for the whole contact interval. After the end of the contact, the object is released, and the method waits for the next contact interval to proceed with the next interaction.

To better distinguish between the objects and improve our robustness to false positive interactions, the contact predictor of iReplica comprises a set of transformers, one for each interaction class: one for the sliding objects and one for the hinged ones. These two

networks work in parallel: when one detects a starting contact, the object search in the neighborhood is performed only for the specific category. The contact is ignored if no object of that class is identified in the user's proximity.

**Adding Hands Data** If fine-grained hand positions are available (*e.g.*, captured with motion capture gloves), we can modify the algorithm to consider such data. Namely, we replace the SMPL model with SMPL+H [RTB17], which has the same template body mesh but provides additional 30 joints (3 joints for each finger) for detailed hand pose representation. We additionally change the input of our contact prediction network to accept vectors of concatenated body pose and hand pose parameters; therefore the input becomes 61 vectors of size $\hat{S} = 159$, adding 90 parameters of hands pose to each input vector. No other architecture changes are made.

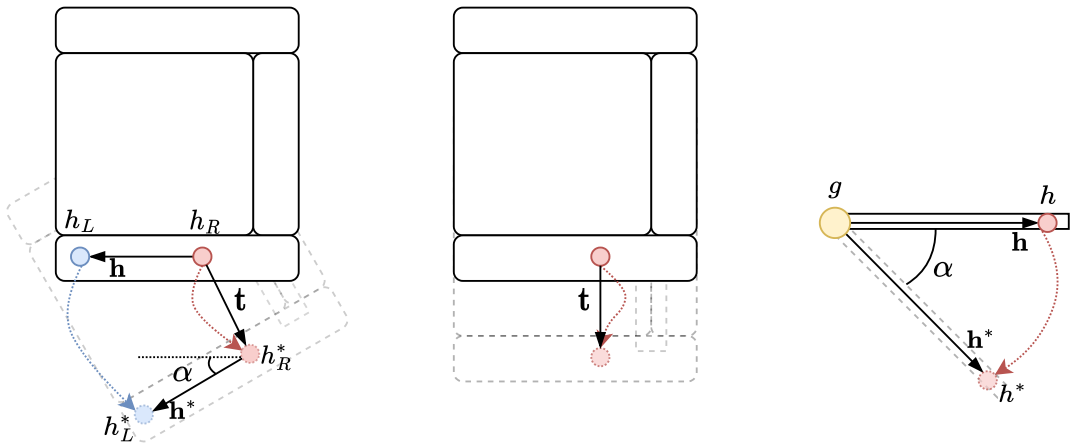
**Training details.** We train the network for 100 epochs with a batch size of 100 using the Adam optimizer [KB15] with a learning rate of $10^{-3}$ and a binary cross-entropy loss. The resulting architecture has 21.9k parameters and an inference time of less than a second per minute of motion (3600 motion windows) on an Nvidia RTX 3090 GPU.

### 5.3.3 Interaction modeling

The benefits of iReplica's pose-based contact prediction and human localization are best visualized by dynamically adapting the scene changes as their consequence. Concretely, when contact with an object is predicted, we attach the object to the user; its dynamic is driven by human motion given through IMU pose and the object's degrees of freedom defined in the interactive environment. Below, we provide more details on the algorithm, also depicted in Fig. 5.7.

**Degrees of freedom.** We model the degrees of freedom (DOF) of the motion of each object to avoid unrealistic motions. For example, a door can only rotate around a specific axis, and a sofa can only slide along the floor. While our model operates in a 3D scene, we present all derivations in 2D since all the processed motions happen along the floor, and any change in the direction orthogonal to the movement plane does not affect the object trajectory and will be removed.

**A) Two-hands interaction with sliding objects.** We consider the point cloud of an interactive object $O \in \mathbb{R}^{n\times 3}$ that can be freely moved on a two-dimensional plane. When the two hands contact the object, we denote the position of two keypoints corresponding to the middle of each hand as $h_L = (x_L, y_L)$ and $h_R = (x_R, y_R)$ for the left and the right hand, respectively, and $\mathbf{h}$ as the vector that connects $h_R$ to $h_L$. When the human moves,




A) Two-hand interaction with sliding objects

B) One-hand interaction with sliding objects

C) Interaction with hinged objects

Figure 5.7: **Obtaining object trajectory from hand interactions**. Colored dashed lines denote hands trajectories, $\alpha$ is inferred rotation angle, $\mathbf{t}$ is inferred object translation vector

we register the new positions of the hands as $h_L^*$ and $h_R^*$, together with the new connecting vector $\mathbf{h}^*$. Then, we compare the hand configurations to recover a translation and a rotation. Firstly, we define the translation $t$ as

$$\mathbf{t} = (x_R^* - x_R, y_R^* - y_R). \tag{5.3}$$

Then, we compute the rotation angle as

$$\alpha = \arccos\left(\frac{\mathbf{h}\mathbf{h}^*}{|\mathbf{h}||\mathbf{h}^*|}\right). \tag{5.4}$$

Finally, we recover the 2D rotation matrix $\mathbf{R}^\alpha$ associated with the angle. The sign of $\alpha$ encodes the direction of the rotation, and it can be obtained by taking the cross-product between the vectors $\mathbf{h}$ and $\mathbf{h}^*$.

**B) One-hand interaction with sliding objects.** When only one hand $h$ interacts with the object, without any further information about the object's physics (*e.g.*, its friction with the ground), it is impossible to recover the object rotation. Hence, given a new configuration $h^*$, we just compute the translation

$$\mathbf{t} = (\widehat{x} - x, \widehat{y} - y). \tag{5.5}$$

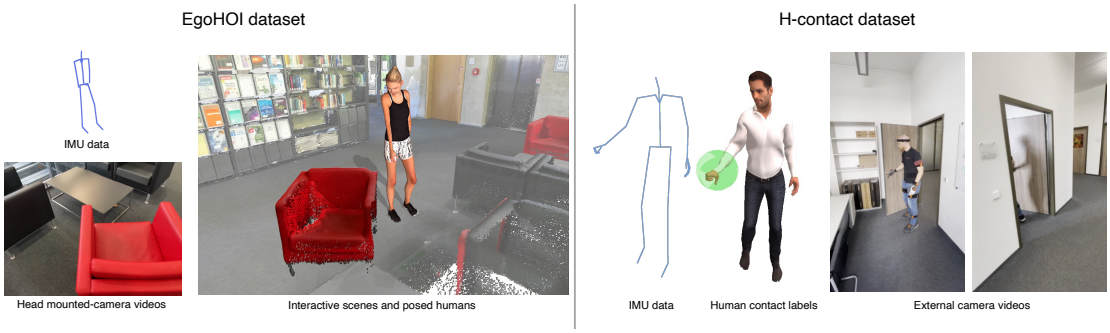



Figure 5.8: **EgoHOI and H-Contact examples.** For EgoHOI, we report for each timestamp the data obtained by the IMUs, the head-mounted camera frame, and the 3D posed human inside the interactive scene. For H-Contact, we provide IMUs data, contact labels for each hand, and recordings from external cameras.

**C) Interaction with hinged objects.** In this case, the object has a hinge positioned in $g = (x_g, y_g)$. When the contact begins at the point $h = (x_h, y_h)$, we compute the vector $\mathbf{h}$ that connects $g$ to $h$:

$$\mathbf{h} = (x_h - x_g, y_h - y_g). \tag{5.6}$$

When the human moves, we register the new position of the contact point $h^*$ and accordingly recompute the connecting vector $\mathbf{h}^*$. Then we compute the angle $\alpha$ between $\mathbf{h}$ and $\mathbf{h}^*$ as in Equation (5.4), and we recover the associated rotation matrix $\mathbf{R}^\alpha$.

After obtaining these transformations, we apply them to the first two coordinates of each point of the object $O$.

$$O^* = \mathbf{R}^\alpha O + t \tag{5.7}$$

## 5.4 Experiments

### 5.4.1 Datasets

In this work we captured and annotated two new datasets: **H-contact** and **EgoHOI**, which we release together with our annotation tool.

**H-contact** is a dataset of human–object interactions, designed to serve as a training set for our contact predictor. We captured and annotated more than 2300 human–object interactions in $>$ 3 hours of recordings divided into 30 uninterrupted sequences. A total of around 680k frames, providing interaction for 8 subjects and 12 objects, whose lengths

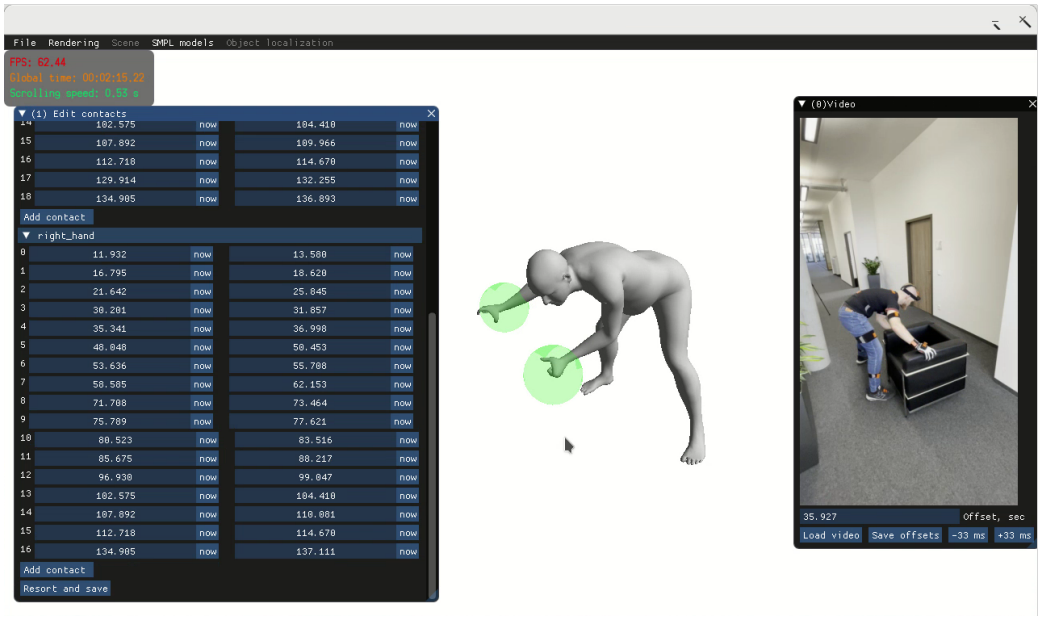



Figure 5.9: **Annotation tool.** The user views the RGB video frames (*right*) and annotates the start and the end of contact interaction (*left*). To help with disambiguating occlusions, the tool also shows the 3D pose (*center*) together with the annotated presence of contact for each hand (green circles).

range from 1 to 19 seconds. To obtain ground-truth contact labels, we built a GUI-based annotation tool for this task. Using synchronized video from an external camera, we asked annotators to define the contact classifications.

**Egocentric Human–Object Interactions (EgoHOI)** is a dataset of humans performing everyday interactions with objects in real scenes recorded with wearable sensors. The sensors are placed directly on the human to allow for large recording volumes not restricted by external camera placement. The wearable setup consists of the IMU-based motion-capturing suit Xsens Awinda [PSRB18], allowing us to obtain human pose sequences, and a head-mounted RGB camera for visual localization of the subject in the scene. The dataset also includes the related interactive environments (IE): a 3D scans of the scene, segmented objects and their degrees of freedom. EgoHOI contains interactions with 14 objects (tables, windows, doors, drawers, sofas, chairs, *etc.*) in multiple IEs for a total of more than 100k motion frames. We also recorded RGBD data from an external multi-camera setup to measure reconstruction accuracy.

**Examples from the Datasets.** In Fig. 5.8 we report some examples from the H-contact and EgoHOI datasets. For H-contact, we provide IMU measurements, SMPL parameters, contact annotations, and external camera recordings. For EgoHOI, we provide interactive

scenes, recordings from head-mounted RGB cameras, IMU data, as well as contact labels and GT final object positions (for evaluation purposes).

**Annotation Tool.** Fig. 5.9 is a screenshot of our annotation tool created for preparing the H-contact dataset. Given a video frame (*right*) and the reconstruction (*center*), the user can annotate contacts by clicking on the interacting hands. The annotator can seek forwards and backward in the video, and the 3D reconstruction helps to disambiguate occluded poses. On average, it takes around 2-4 seconds to annotate 1 second of video.

### 5.4.2 Implementation and Performance

We implement our algorithms in Python using the PyTorch [PGM+19] library for the contact prediction network and the bending energy optimization algorithm. For the latter, we use the Adam [KB14] optimizer with 1000 iterations, with the learning rate and the rigidness coefficient $\lambda$ acting as hyperparameters.

The most computationally expensive part of our algorithm is the visual localization pipeline needed for HPS, requiring 8 seconds per frame on an NVIDIA Q8000 GPU, while the other steps take little additional time. Such a computationally expensive pipeline provides good localization results and supports our exploration. Note that the performance of the visual localization network itself is not the focus of our study, and we expect that more efficient alternatives can replace this algorithm in the future.

### 5.4.3 Baselines

Due to the novelty of the proposed human-object tracking task, no published baselines exist. Therefore, we introduce novel baselines and describe them below.

**HPS.** We compare to HPS [GMSPM21] that localizes the human within the prescanned scene using the images of the head-mounted camera. HPS does not reason about human–object interactions and does not track scene changes.

**HPS w/ GT** combines HPS with ground-truth data to predict the object's motion. In this baseline, we assume that an oracle provides the ground-truth final object position, as well as the time window of the interaction. We stress that neither of these is available at inference time in our setting, and our method does not rely on them. Then the human motion is solely estimated using HPS, and the object movement is modeled using linear interpolation for translation and spherical linear interpolation (Slerp [Sho85]) for rotation. By inspecting the qualitative results, we see that motion between humans and objects happens asynchronously and, therefore, unrealistically. This baseline shows that, even in the

presence of further assumptions, modelling the object trajectory is non-trivial. It is clear that human motion is rarely linear, and more sophisticated techniques are required.

**HPS w/ RGB Obj. Loc.** localizes the object using solely the RGB frame from the head-mounted camera. In this baseline, we assume that for each frame of the head-mounted camera, we have a perfect 2D segmentation mask for the object, obtained by semi-automatic annotation using an interactive segmentation pipeline [SPBK20]. Then we use the same localization method [SCSD19] as HPS to provide a 6-DoF localization of the object w.r.t. the human. Starting from the dataset of images with the known object positions in the pre-scanned scene, the algorithm establishes correspondences between those images and frames from the head-mounted camera. The head-mounted camera is then localized by minimizing reprojection loss. Next, the object's location relative to the camera is recovered by matching and optimizing with only the 2D key points inside the object mask. This information is combined with the camera position to recover the object's location in world coordinates. This baseline highlights that the camera is unreliable for localizing the object in the space. Detecting local landmarks is dramatically harmed by occlusions, head shaking, and the object missing in several frames.

**iReplica w/ HOD.** Given the availability of a head-mounted camera, we explore the possibility of using it to predict contacts. In this baseline, we replace our pose-based contact predictor with HOD [SGSF20], a method trained on many YouTube videos. Starting from a single RGB image, it predicts a full set of hand interaction properties: hands bounding box, object bounding box, and the contact state for each hand. We keep the rest of our method fixed except for the contact prediction part. The original paper [SGSF20] shows several results from an egocentric perspective, but it also mentions failure cases when hands and objects are close to each other. We confirm this by qualitative inspection, noting that the method loses contact with the object during the interaction.

**iReplica w/ VISOR.** Given that HOD is trained on a variety of videos, which also include extrinsic views, we deploy a similar baseline but rely on an RGB method specifically trained on egocentric views. Specifically, we consider the baseline trained for the HOS challenge[3] on the VISOR [DSZ+22] dataset. This baseline relies on PointRend [KWHG20] segmentation method, augmented with auxiliary detection heads to predict the contact, following the same idea as HOD [SGSF20]. As in the previous baseline, we replace our pose-based contact prediction, leaving other parts of the method untouched. However, also in this case, we observe missing contacts and wrong release prediction, often causing human penetration (*e.g.*, crossing the door).

[3] https://github.com/epic-kitchens/VISOR-HOS


Figure 5.10: **Qualitative results.** We show three examples of human interaction, pairing the head-mounted camera view with the interaction modeling achieved by iReplica. The object is not always visible during the interaction (Interaction 1), hand grasping can be difficult to understand from the camera (Interaction 2), or object occludes a majority of the first person view (Interaction 3). By relying on human-centric contact detection, iReplica achieves reliable modeling in all these challenging scenarios. Please see our video for more results.

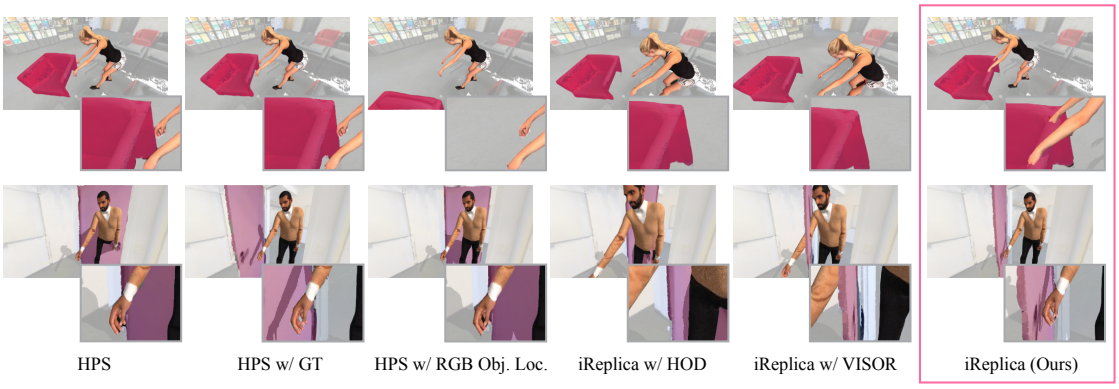



Figure 5.11: **Qualitative comparison.** We compare iReplica (ours) to the baseline methods (interacted object highlighted in red for visual clarity). For the sofa sequence (top row) no baseline can track the sofa and correctly place the subjects‘ hands. Similarly, the door (bottom row) is incorrectly placed by all baselines, and the hand is not in contact with the handle. In contrast, iReplica obtains visually plausible results by adjusting human and object locations to satisfy contact constraints.

### 5.4.4 Results

**Qualitative results.** Fig. 5.10 shows our results for sample frames from multiple-scene interactions. Our results show that egocentric motion data alone can localize the human in the scene, model the interaction between the human and objects, and update the scene accordingly. Our contact predictor allows iReplica to estimate object tracking only based on the human pose; *e.g.*, doors, chairs, and windows can be interacted with in the scene.

Fig. 5.12 highlights the features of iReplica on the sequence, recorded to simulate a realistic and natural scenario. iReplica does not assume a single object interaction: instead, it applies to interactive scenes where many objects can be differently re-arranged by the interaction (*e.g.*, table, chair). iReplica also works with objects with non-linear motion in space, for example, the doors that might start and end their movement in the same place. In this case, for example, an interpolation baseline would not provide any object dynamic since the initial and final configurations are the same. Instead, iReplica tracks the full trajectory of the object. iReplica allows the user to move in the space freely and interact naturally as in everyday life.

**Qualitative comparisons to baselines.** Fig. 5.11 visually compares iReplica to our baselines by showing individual frames from some of the interactions.

*HPS* does not track scene changes and thus obtains unrealistic motions. For example, the door opening is not tracked. The subject should have opened the door with the handle,

| Error ↓ | Method | Door | Sofa | Table | Box | All |
|---|---|---|---|---|---|---|
| Distance (in cm) | **HPS** | 79.27 | 69.54 | 25.31 | 41.92 | 60.81 |
| | **HPS w/ RGB Obj. Loc.** | 28.66 | 1684.06 | 119.59 | — | 597.83 |
| | **iReplica w/ HOD [SGSF20]** | 57.50 | 55.78 | 1.62 | **3.33** | 38.58 |
| | **iReplica w/ VISOR [DSZ+22]** | 43.40 | 66.74 | 5.98 | 11.31 | 39.59 |
| | **iReplica w/o Contact corr.** | 18.54 | 11.70 | 1.84 | 7.79 | 11.68 |
| | **iReplica (Ours)** | **9.97** | **6.66** | **0.90** | 7.09 | **6.88** |
| Angle (in °) | **HPS** | 109.19 | 23.53 | 12.16 | 3.76 | 46.89 |
| | **HPS w/ RGB Obj. Loc.** | 34.36 | 118.02 | 60.08 | — | 61.43 |
| | **iReplica w/ HOD [SGSF20]** | 75.74 | 7.74 | 0.78 | **2.71** | 28.41 |
| | **iReplica w/ VISOR [DSZ+22]** | 56.64 | 17.36 | 2.87 | 12.78 | 27.27 |
| | **iReplica w/o Contact corr.** | 22.16 | **5.83** | 0.88 | 4.81 | 10.28 |
| | **iReplica (Ours)** | **12.94** | **5.83** | **0.43** | 4.81 | **7.13** |

Table 5.1: **Object localization accuracy.** Distance (in cm) and angle (in °) between object center at the end of the interaction in the GT pose and object center in the pose predicted by the algorithm.

| Method | Door | Sofa | Table | Box | All |
|---|---|---|---|---|---|
| **HPS** | 46.00 | 38.32 | 26.35 | 6.64 | 33.61 |
| **HPS w/ GT** | 17.28 | 6.90 | 7.55 | 6.74 | 10.44 |
| **HPS w/ RGB Obj. Loc.** | 65.26 | 724.63 | 136.27 | — | 287.12 |
| **iReplica w/ HOD [SGSF20]** | 48.42 | 35.96 | 13.31 | 5.52 | 31.26 |
| **iReplica w/ VISOR [DSZ+22]** | 33.76 | 51.14 | 13.39 | **3.84** | 31.17 |
| **iReplica w/o Contact corr.** | 18.15 | 9.80 | 6.89 | 5.45 | 11.37 |
| **iReplica (Ours)** | **2.83** | **1.46** | **3.49** | 5.49 | **2.93** |

Table 5.2: **Visual plausibility of human-scene interaction.** Mean distance between the object and the contacting hand (in cm) over the interaction time.

| Contact predictor | AP ↑ | Precision@0.5 ↑ | Recall@0.5 ↑ | Accuracy@0.5 ↑ |
|---|---|---|---|---|
| **HOD [SGSF20]** | 0.044 | 0.251 | 0.818 | 0.364 |
| **VISOR [DSZ+22]** | 0.217 | 0.313 | 0.098 | 0.732 |
| **POSA [HGT+21]** | 0.033 | 0.115 | 0.716 | 0.297 |
| **Ours** | **0.807** | **0.786** | **0.880** | **0.905** |

Table 5.3: **Contact prediction performance.** Metrics obtained on our test set with subjects that are not appearing in training data.

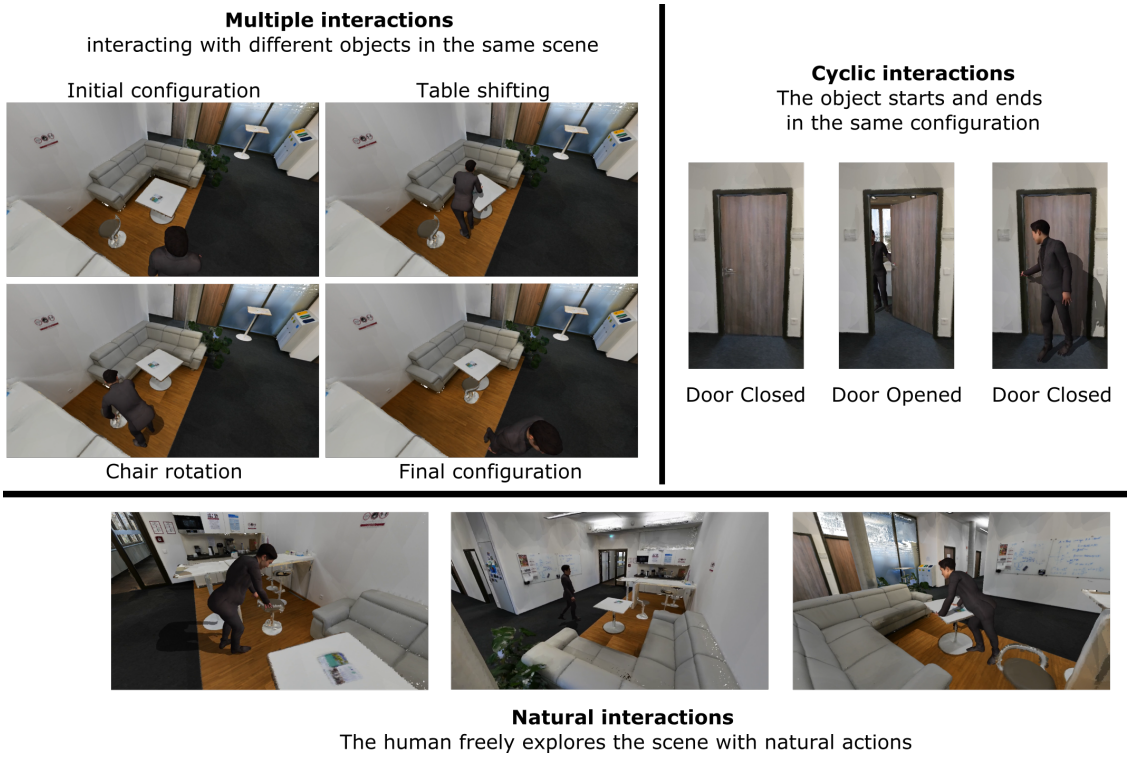



Figure 5.12: **Features of iReplica.** iReplica handles different kinds of challenges. It can be naturally applied to interactions involving several objects (*e.g.*, arranging a table and a chair), objects that have a cyclic behavior in the scene (*e.g.*, doors that open and close several times), and general daily interactions where the user freely moves in the space (*e.g.*, walking to the kitchen, pushing a table).

but the door is still closed. (Fig. 5.11, door). Or the sofa is dragged by the subject, but the object stands still. The sofa, therefore, is visibly not in contact with the subject's hands (Fig. 5.11, sofa).

*HPS w/ GT* linearly interpolates the object motion given ground-truth object start and end pose and contact times. The resulting interaction is not visually plausible due to a large mismatch between the subject's hands and the sofa. Human-scene interaction motion is highly non-linear, so linear approximations seem unrealistic; this is also proven by the user study below.

*HPS w/ Obj. Loc.* fails to detect the accurate object position during the interaction, resulting in misaligned results, to the point that it fails to localize the object at all (*e.g.*, Tab. 5.2, Box).

*iReplica w/ HOD* and *iReplica w/ VISOR* both suffer from false negatives, resulting in a sudden contact loss in the middle of the interaction and missed contacts between object

and subject.

*iReplica* ensures that the subject's hands are close to the object during the whole interaction – a key aspect of visual plausibility not achieved by the baselines. This shows the value of iReplica's correction of the human trajectory based on the human–object interaction.

**Reconstruction quality compared to real scenes.** We quantitatively validate iReplica's object and human localization results in terms of the reconstruction quality with respect to the original scene. We measure deviations from the virtual replica to the real scene using the EgoHOI dataset. Tab. 5.1 shows the object localization accuracy at the end of the interaction, where the GT object pose was annotated. On average, iReplica improves the results considerably (col. *All*). All object types are localized with a distance below 10 cm and an orientation error below 13 degrees.

**Ablation of Contact-based human trajectory correction.** We ablate the contact-based human trajectory correction by excluding it from iReplica. We report the results in Tables 5.1 and 5.2 (**iReplica w/o Contact corr.**). The method greatly benefits from the proposed correction.

**Visual plausibility.** To measure the visual plausibility of iReplica results compared to the baselines, we consider the contact between the human and the object. In particular, we measure the mean distance from the object to the interacting hand, see Tab. 5.2. iReplica keeps this distance below 3 cm. Tracking contacts and using them for attaching the object to the human motion creates the lowest distances.

**Contact prediction accuracy.** We benchmark the accuracy of iReplica contact prediction in isolation in Tab. 5.3, comparing it to our two RGB contact prediction baselines, HOD [SGSF20] and VISOR [DSZ+22]. We treat the network predictions as probabilities in a binary classification task and compute 4 metrics: Average Precision (AP), Precision, Recall and Accuracy on the binarization threshold of 0.5.

Our contact prediction, solely based on the 3D human pose, significantly outperforms the RGB-based reasoning - one cause is that interaction is not always visible in the camera. Once more, we remark on how 3D human poses in isolation is a highly informative indicator of interaction contacts.

In Tab. 5.3 we additionally present a comparison with a method of contact detection appearing in POSA [HGT+21]. POSA presents a human-scene interaction model that can be used as a prior for human placement in the scene. Compared to iReplica, POSA has a different goal and applications: it does not consider dynamic interactions, works only with one human pose frame at a time and is focused on static scenes. However, this

| $t$ | Error (cm) |
|---|---|
| 0.00 | 13.61 |
| 0.25 | 8.49 |
| **0.50** | **7.13** |
| 1.00 | 8.86 |
| 2.00 | 8.86 |

Table 5.4: **Contact interval study.** The maximum gap to fill between two active contacts (in seconds) and the resulting positioning error of iReplica on the validation set (for EgoHOI dataset).

is the closest baseline for our task of temporal pose-based interaction prediction. For a fairer comparison, we finetune the POSA model on the same training subset from the H-Contact dataset used for iReplica; we also average contact prediction scores from a selected region of each hand and treat this as a per-hand contact probability.

The comparison shows that the POSA model could not estimate contacts reliably, even after finetuning it on our training data. One possible reason for this could be that this method works with single frames, which leads to prediction uncertainty for many poses, while iReplica's transformer-based model considers a one-second window of motion.

**Contact interval study.** To remove false negatives, we post-process the predicted sequence and fill in gaps between active contacts that are shorter than a certain threshold $t$. To determine the optimal value of $t$, we conducted an experiment on a validation set of EgoHOI. The results are reported in Table 5.4. We found that, according to the experiments, value of 0.5 s performs the best.

**User study.** To measure the realism of the motion produced by linear interpolation, we did a user study and asked 73 respondents to rank the interactions produced by iReplica, HPS w/ GT and HPS by realism. Each participant was given 9 questions, each showing results generated by two methods side-by-side in a random sequence. iReplica results were preferred in 84.2% of the cases, proving that our body-driven object tracking produces more realistic interaction.

**External Camera Evaluation.** To additionally measure the human-object localization accuracy of our method, we recorded a special sequence with ground-truth data obtained via an external multi-view system of 3 depth cameras. The experimental setup closely follows the one used in HPS, however we capture the full dynamic object interaction while in HPS only the human body motion and the static scene was captured.

We use 3 calibrated Azure Kinect [Mic24] RGBD sensors. By combining the outputs

| | Human to GT $E_{body} \downarrow$ | Object to GT $E_{obj} \downarrow$ |
|---|---|---|
| **HPS** | 9.771 | 22.651 |
| **iReplica (ours)** | **8.981** | **8.471** |

Table 5.5: **Human and object tracking quality w.r.t. the real scene:** Mean 3D error (in cm) between tracked and moving human/object models compared to the real scene. The scene is captured via a synchronized, multi-view RGBD video recording setup observing the interaction.

of these sensors, we obtain a sequence of 3D point clouds of the scene and a subject. The Azure Kinect features built-in temporal synchronization, but to merge the output of the sensor into the scene ground truth representation, we also need to calibrate them spatially. For that we use a three-stage localization pipeline, similar to HPS in Sec. 4.5.1. Using the obtained positions of the sensors, the point cloud representation of the scene is formed by unprojecting depth maps from all 3 sensors to 3D. To perform the evaluation, we manually synchronize the time between the iReplica motion sequence and the aforementioned point cloud representation. For each frame of the test sequence, we separately measure object and human localization error $E_{obj}$ and $E_{body}$:

- $E_{obj}$: mean Chamfer distance from the object point cloud to the ground-truth point cloud,
- $E_{body}$: mean Chamfer distance from the human body SMPL mesh to ground-truth point cloud.

Results are presented in Table 5.5. Since HPS models only humans, the large error from the object ground truth is not surprising. Although HPS and iReplica share the same camera localization principle, we observe that our iReplica improves human localization. This validates that using the detected human–object interaction to adapt the human localization trajectory helps to improve reconstruction correctness. Moreover, it drastically enhances visual plausibility, as explained in the qualitative analysis.

## 5.5 Discussion and Conclusion

In this chapter, we proposed the novel problem of capturing human-scene interactions and dynamic 3D scenes solely from wearable sensors - that is, IMUs and a head-mounted camera, and not relying on any external cameras or object trackers. We show that egocentric data alone can be used to localize the human in the scene, model the interaction

with the objects, and update the scene accordingly. iReplica enhances results of human localization from HPS, correcting that the human and the interacted object are close to each other.

**Future work and Limitations.** Our simple method has some natural limitations, which point to interesting future directions. Our approach does not consider physics or collisions, which future work can investigate using simulations similar to those available in game engines. Our work relies on a prescanned IE, in which objects degrees of freedom are known a priori. Incorporating environment reconstruction (e.g., SLAM-based models) and on-the-fly segmentation (e.g., ScanNet) could remove these assumptions. Finally, our paradigm does not consider more complex interactions and object manipulations (e.g., articulated, non-rigid). This would be an exciting research direction, especially in light of the recent availability of human-object datasets [FTT+23].

# Chapter 6

# Human Motion Diffusion from Head-Mounted Device (HMD$^2$)

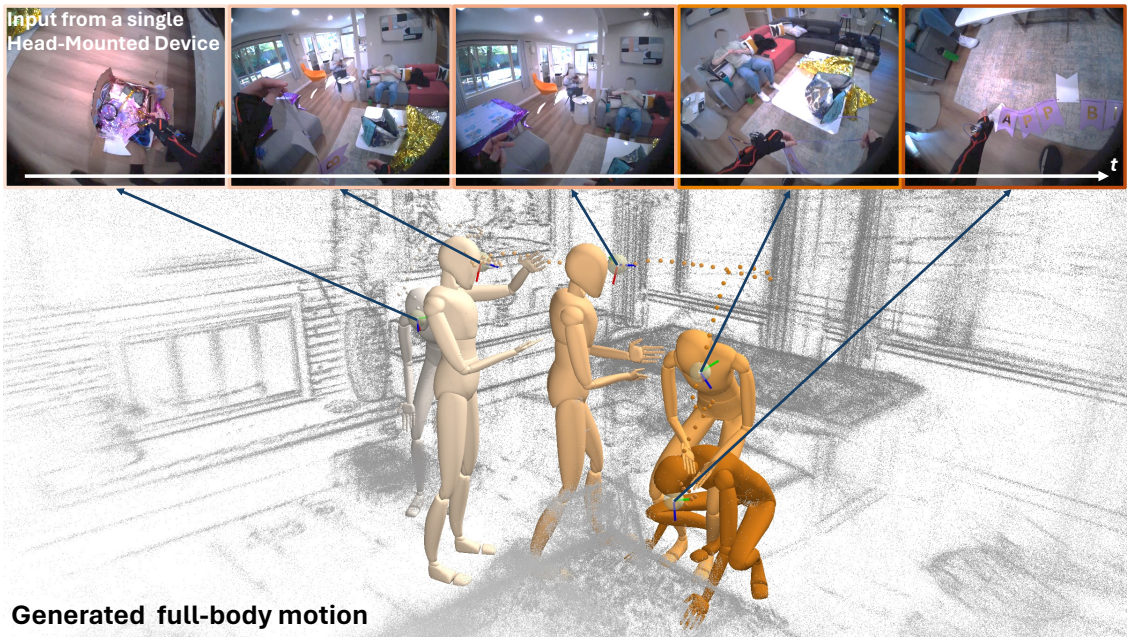



Figure 6.1: We propose HMD$^2$, the first system for the online generation of full-body motion using a single head-mounted device (*e.g.* Project Aria Glasses) equipped with an outward-facing camera in complex and diverse environments.

This chapter investigates the generation of realistic full-body human motion using a single head-mounted device with an outward-facing color camera and the ability to perform visual SLAM. To address the ambiguity of this setup, we present HMD$^2$, a novel system that balances motion reconstruction and generation. From a reconstruction standpoint, it aims to maximally utilize the camera streams to produce both analytical and

learned features, including head motion, SLAM point cloud, and image embeddings. On the generative front, HMD$^2$ employs a multi-modal conditional motion diffusion model with a Transformer backbone to maintain temporal coherence of generated motions, and utilizes autoregressive inpainting to facilitate online motion inference with minimal latency (0.17 seconds). We show that our system provides an effective and robust solution that scales to a diverse dataset of over 200 hours of motion in complex indoor and outdoor environments.

This chapter is based on [GJH+25][1] led by Vladimir Guzov jointly with Yifeng Jiang, with equal contribution: Vladimir Guzov's contributions are autoregressive motion inference, image conditioning module and motion postprocessing algorithms, while Yifeng Jiang contributed with scene conditioning module and motion preprocessing algorithms. Both authors contributed equally to diffusion architecture design and method evaluation.

## 6.1 Introduction

Wearable devices such as smart glasses promise to become the cornerstone of next-generation personal computing. A key challenge is accurately interpreting the wearer's motion from the device's limited input signals, taking into account the social and environmental context at the moment. The capability to generate full-body movements solely from a single head-mounted device (HMD) in real-time, outdoors and indoors, will open the door to many downstream applications, including telepresence, fitness and health monitoring, and navigation.

State-of-the-art methods, such as EgoEgo [LLW23], have shown visually impressive results in a similar context. However, these systems operate offline, are optimized for generating short windows of motion, and are mostly trained on a small set of indoor motions. More crucially, they utilize the head-mounted camera only for head pose estimation, missing the opportunity to harness additional image features of the environment and of the wearer's own body.

We introduce HMD$^2$ (Human Motion Diffusion from Head-Mounted Device), the first system, to our knowledge, capable of online generation of full-body movements from a single HMD (Project Aria Glasses [ESG+23]), conditioned on outward-facing egocentric camera streams in diverse environments. Given that such devices provide limited observation of the body and surroundings, the critical question is how to maximally utilize the input. Our approach reuses input data to generate features across different modalities,

[1] © 2025 IEEE. Reprinted, with permission, from [GJH+25]

covering independent aspects of the environment and motion. Specifically, from the input streams, we mix and match analytical and learning toolboxes to extract 1) wearer's head motion from off-the-shelf real-time visual SLAM; 2) environment feature points as a by-product of SLAM, important for motion disambiguation in complex scenes; and 3) head camera image embeddings (*e.g.* using CLIP [RKH+21]) for additional scene clues and intermittently visible body parts.

However, full recovery of the wearer's motion is still highly under-constrained, given our input. Our system takes a generative approach and adopts a diffusion-based Transformer backbone to strike a balance between motion reconstruction and motion generation. This allows for the generation of diverse outcomes, such as varying leg movements, from identical inputs. Additionally, our diffusion model can predict motions with minimal future sensor information (0.17 seconds), facilitating online and real-time use cases.

Contrary to evaluations using large synthetic datasets or small-scale real-world datasets, we train and test our system on the extensive 200-hour real-world Nymeria dataset [MYH+24] recorded with publicly available head-mounted device, containing various indoor and outdoor activities performed by over 100 subjects with diverse body sizes and demographics. While most existing research on motion tracking is evaluated solely based on reconstruction accuracy, we acknowledge the inherent ambiguity in our problem and evaluate our system on generation fidelity and diversity as well. Our contributions are summarized as follows:

1. We present a novel application of online full-body motion generation from a single HMD. The multi-modal feature streams extracted from the device serve as a key ingredient for the system's success across a diverse set of environments.

2. We employ a multi-modal conditional motion diffusion backbone, effectively balancing between accurate motion reconstruction and the diversity and fidelity of synthesized movements.

3. We demonstrate the adaption of a time-series motion diffusion model for online autoregressive inference through inpainting, eliminating the dependency on future sensor input and achieving minimal latency.

4. We evaluate the proposed system with large-scale, real-world Nymeria [MYH+24] dataset and achieve state-of-the-art performance for single-HMD motion generation.

## 6.2 Method

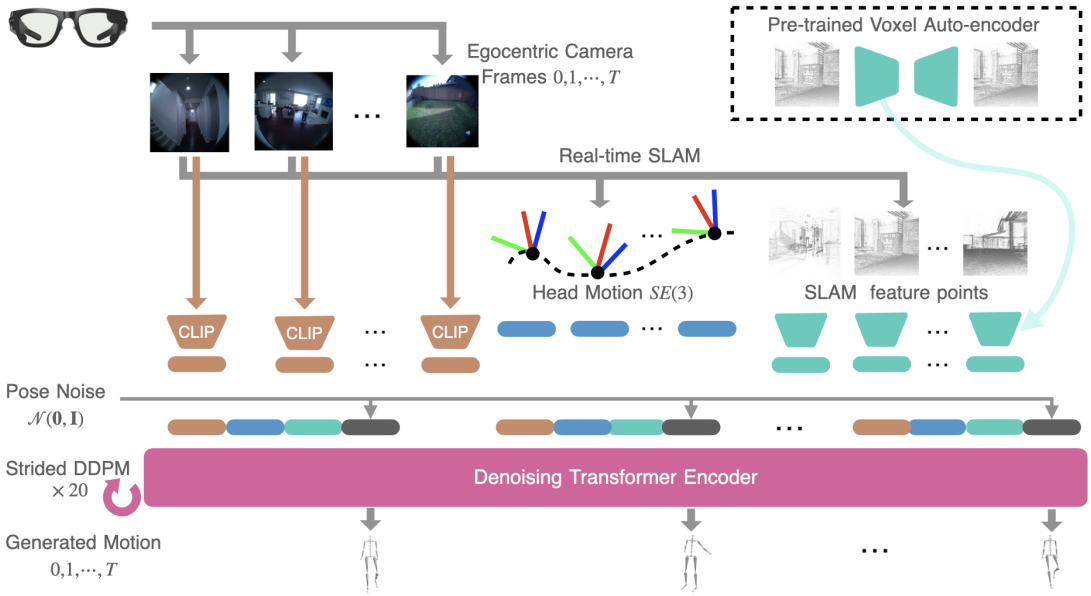



Figure 6.2: Overview: HMD$^2$ generates realistic full-body motion that aligns with the signals from a single head-mounted device. Using the image streams from the egocentric camera and head trajectory with the feature cloud from the onboard SLAM system, we employ a diffusion-based framework to generate the wearer's full-body motion.

We introduce a diffusion-based framework for generating full-body motion based on multi-modal signals from an HMD, like the Project Aria Glasses [ESG+23]. As shown in Fig. 6.2, our system uses device with an outward-facing camera, capable of real-time SLAM [Pro23] (which may utilize other sensors) which produces a 6D pose trajectory, and a spatial map of the environment represented by an aggregated point cloud. We extract contextual information from both the environment point clouds and the egocentric video stream, using a CLIP encoder [RKH+21] for image embedding and an independently trained point cloud autoencoder for spatial map embedding to supplement the 6D pose.

Given the under-constrained nature of the task, we employ a diffusion model [HJA20] with a time-series Transformer encoder [VSP+17] to model the motion distribution. To ensure temporal consistency during streaming, we use autoregressive inpainting during denoising, aligning new body motion with previous predictions.

Figure 6.3: A typical input sequence from egocentric camera with only few body parts of the wearer intermittently visible, rendering standard full-body reconstruction network backbones ineffective.

### 6.2.1 Multi-modal Scene and Motion Conditions

Our model is trained to align its output with three modalities of features, all of which are streaming frame by frame to allow infinitely long motion generation. For each frame, the inputs include a head pose $(\boldsymbol{t}, \boldsymbol{R}) \in \mathrm{SE}(3)$ representing the head's position and orientation, a color image $\boldsymbol{I}$ from the camera, and a set of SLAM feature points $\boldsymbol{S} \in \mathbb{R}^{N\times 3}$ of the surrounding scene. We concatenate features per-frame and process the resulting vector with a linear layer. We elaborate on each modality and their respective design considerations below.

**Head Pose Trajectory.** The device pose provides precise spatial location and movement of the wearer's head. We augment the device pose vector with its linear and angular velocity vector $(\boldsymbol{v}, \boldsymbol{\omega})$ computed from finite differences to form $\boldsymbol{p} = \{\boldsymbol{t}, \boldsymbol{R}, \boldsymbol{v}, \boldsymbol{\omega}\}$. We canonicalize each window of $\{\boldsymbol{p}\}_{0,1,\cdots,T}$ to its first frame $\boldsymbol{p}_0$, allowing the model to function in arbitrarily large spaces and generate infinitely long sequences. This is crucial for navigation in a multistory building or outdoor hiking with large elevation changes.

**Camera Image Embeddings.** Beyond the head pose trajectory derived from visual SLAM, the egocentric camera images offer additional valuable information. For example, when a body part becomes visible, the image provides a strong cue of the wearer's pose. However, direct utilization of the image content proves less useful, as it may capture distracting texture details when all we need is high-level semantics such as "the left hand is above the waist." Empirically, we found that CLIP embeddings [RKH+21], $E_I(\boldsymbol{I})$, provide a significant performance boost to the learning process while avoiding overfitting to superficial image characteristics.

It is important to emphasize that embeddings provided by human-related backbones, such as networks trained for pose reconstruction from monocular videos, will not work well in our case. Figure 6.3 shows a typical input camera sequence when parts of the self-

body (hands in this example) are visible. This differs significantly from downward-facing egocentric cameras, which observe most of the body. This discrepancy leads to failures in existing network backbones for full-body motion, and it may be tempting to assume that such input might not be useful for full-body motion reconstruction. However, high-level descriptions of the images that contain scene information, such as "hand reaching to the sink" (which is typically at a standard height) or "a person kicking a football (implicitly indicating that the wearer might also soon interact with the ball)", are actually quite useful for spatial reasoning of the wearer's end effectors. We hypothesize that this observation explains why CLIP embeddings are advantageous in our unique problem setting.

**SLAM Point Cloud Embeddings.** Visual SLAM algorithms identify static feature points in the environment (*e.g.* corners and edges of furniture) and aggregate them over time to build 3D maps. These points offer crucial environment features to constrain motion generation, akin to pre-scanned scenes utilized in prior work [GMSPM21, LJ24]. At each frame, we only consider the available SLAM feature points $\boldsymbol{S}$ within a 2m $\times$ 2m $\times$ 2m volume. The center of the volume is the current device position offset downwards by one meter, similar to prior works [SZKS19]. This ensures the model focuses only on relevant spatial information as the wearer moves around. The points are voxelized in a $10 \times 10 \times 10$ voxel grid in the following way: for each voxel center, the closest point is selected and the distance is stored as a voxel value. All the distances are truncated at 10cm (so the value is clipped between 0 and 0.1). The voxel volume is rotated with the head orientation but only along the Z (gravity) axis. To better handle the noisy and often incomplete nature of SLAM point clouds, we pre-train an autoencoder on the voxelized SLAM point clouds $V(\boldsymbol{S})$ within the bounding volume on all frames in our training dataset and use its encoder $E_S(\cdot)$ to generate point cloud embeddings $E_S(V(\boldsymbol{S}))$. While a new map may not offer much information right away, rich point cloud features could quickly build up if the wearer stays in the same environment for a prolonged period (*e.g.* 15 min) or if they have access to a prebuilt map.

### 6.2.2 Conditional Motion Diffusion Model

Given all input signals from the device, $\boldsymbol{c} = \{\boldsymbol{p}, E_I(\boldsymbol{I}), E_S(V(\boldsymbol{S}))\}_{0,1,\cdots,T}$, diffusion models such as DDPM [HJA20] can model the distribution of all motions conditioned on $\boldsymbol{c}$ by progressively introducing distortions (Gaussian diffusion noises) into the motion sequence and learning a neural network model $D$ to reverse these distortions. The sequence of forward distortions can be described by the following equation:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0,\boldsymbol{c}) = \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_0,(1-\alpha_t)\mathbb{I}) = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \tag{6.1}$$

where the motion $\boldsymbol{x} \in \mathbb{R}^{T\times F}$ is represented as a time series with window length $T$ and motion feature dimension denoted as $F$. Here, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbb{I})$ denotes the unit Gaussian noise, and $t \in \{0,1,\cdots,S\}$ signifies the level of distortion, with $t=0$ indicating no distortion and $t=S$ representing maximum distortion such that $\alpha_S = 0$ and $\boldsymbol{x}_S \sim \mathcal{N}(\mathbf{0},\mathbb{I})$. The parameter $\alpha_t$ is a monotonically decreasing scalar that governs the noise schedule. The reverse diffusion process is derived using Bayes' rule and can be expressed as:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0,\boldsymbol{c}) = \mathcal{N}(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + c_t\frac{(\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_0)}{\sqrt{1-\alpha_t}},\sigma_t^2\mathbb{I}), \tag{6.2}$$

$$c_t = \sqrt{1-\alpha_{t-1}-\sigma_t^2}, \quad \sigma_t^2 = (1-\frac{\alpha_t}{\alpha_{t-1}})\frac{1-\alpha_{t-1}}{1-\alpha_t}. \tag{6.3}$$

With $\boldsymbol{x}_0$ in Eq. (6.2) estimated by the neural net module $\hat{\boldsymbol{x}}_0 = D(\boldsymbol{x}_t,\boldsymbol{c},t)$, we can iteratively generate a sequence of samples $(\boldsymbol{x}_S,\boldsymbol{x}_{S-1},\ldots,\boldsymbol{x}_1,\boldsymbol{x}_0)$, initiating from $\boldsymbol{x}_S \sim \mathcal{N}(\mathbf{0},\mathbb{I})$ and progressing towards the desired motion distribution $q(\boldsymbol{x}_0|\boldsymbol{c})$ over $S$ reverse diffusion steps. During model training, we randomly sample $t$ from a uniform distribution $U(0,S)$ for every training data. At inference time, we apply $\bar{S}=20$ evenly spaced strided reverse diffusion steps [ND21]. Note that no Gaussian noise is applied to the condition vector $\boldsymbol{c}$. Training loss is defined as:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}_0\times t\sim U(0,S)}||D(\boldsymbol{x}_t,\boldsymbol{c},t)-\boldsymbol{x}_0||^2, \tag{6.4}$$

We did not find it necessary to include auxiliary loss terms to refine output quality.

**Architecture and motion inference.** Our conditional motion diffusion model follows the Transformer-based architectures presented in EDGE [TCL23] and DiT [PX23] with additional MLP encoder layers to gradually reduce the input dimension (which is bigger due to added CLIP and PC features) to the token latent space size. Our input consists of the motion input (as a translation, rotation, and linear and angular velocities) and PC and CLIP features, all concatenated together, representing one sequence token per frame. Following AvatarPoser [JSQ+22], the model only predicts local joint rotations but not global translation. The global movement of the character is created during test time by "stitching" the predicted body motion to the ground-truth head motion, and the head motion can be directly obtained through real HMD motion obtained through SLAM, offset by a constant calibration matrix provided by the dataset. The motion output of the diffusion model is denoted as $\boldsymbol{x} \in \mathbb{R}^{T\times F}$, where $T=240$ denotes the prediction window

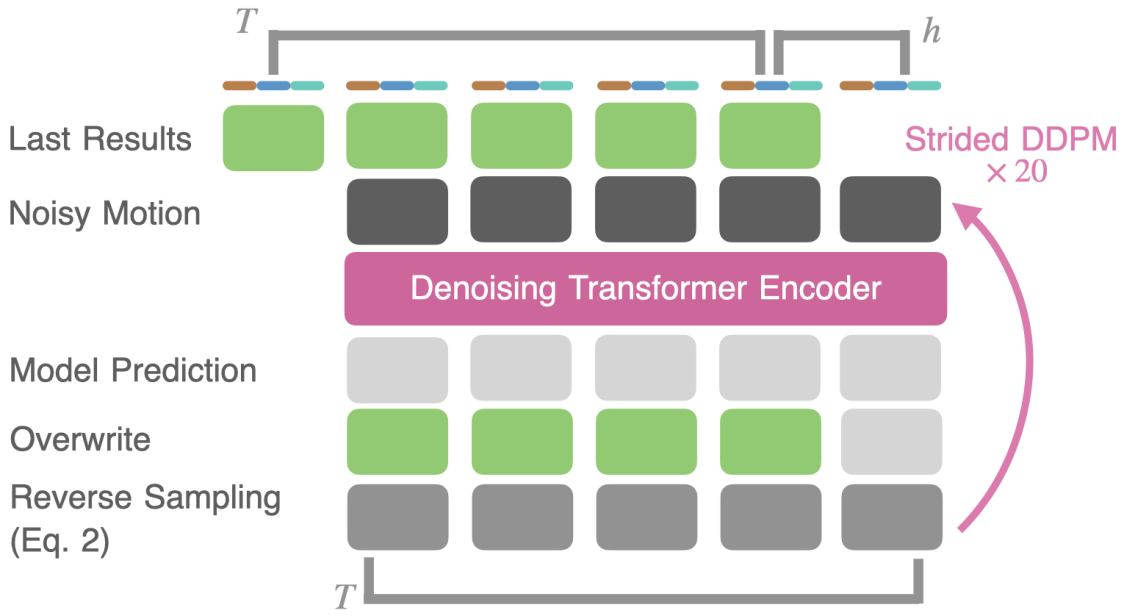



Figure 6.4: Autoregressive inpainting is performed at each reverse diffusion step to allow long sequence generations both in high- and low-latency settings.

size and $F = 23 \times 6$ denotes the size of the single body pose vector. The body skeleton is following Xsens definition (Sec. 2.1.2) and has 23 ball-and-socket joints. For each joint, the output rotation is represented as the first two columns of its local rotation matrix. Note that the definition of Xsens human skeleton is very similar to SMPL [LMR+15], with the main difference being the ordering of joints. The model is not conditioned on body size information, but during training, it is forced to see HMD input motions from different subjects covering highly diverse demographics. As such, the trained model is able to handle body size variation implicitly. However, providing size information as an explicit condition might further improve model performance and reduce visual artifacts such as floor penetration and foot sliding. To create the motion visualizations and compute position error metrics, we used ground truth body sizes (skeleton bone lengths) for each subject.

**Online Inference of Long Sequences.** Our motion diffusion model generates up to 4 seconds of motion ($T = 240$ frames). To extend this for longer, coherent motions, previous research [CEJ+23, ZSH+23, STKB23b] suggests generating overlapping windows and enforcing consistency at overlaps during denoising. However, for online generation, we need to remove the dependency on future windows by using an autoregressive approach [HSG+22], where each window depends only on the previous one.

Specifically, when two windows overlap by $T - h$ frames (i.e., the current window

advances by a stride of $h$), we enforce consistency during each of the $\bar{S}$ denoising steps. After each model evaluation $\hat{\boldsymbol{x}}_0 = D(\boldsymbol{x}_t, \boldsymbol{c}, \tau_i)$, the prediction $\hat{\boldsymbol{x}}_0$ is overwritten by the overlapping prediction from the preceding window:

$$\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 \odot \boldsymbol{m} + \hat{\boldsymbol{x}}_{s0} \odot (1 - \boldsymbol{m}), \tag{6.5}$$

where $\boldsymbol{m} \in \mathbb{R}^{T \times F}$ is a constant mask that is zero for the initial $T - h$ frames and one for the last $h$ frames. $\hat{\boldsymbol{x}}_{s0} = \text{cat}(\boldsymbol{x}_0^-[h{:}T], 0^{h \times F})$ denotes the prediction from the previous window, shifted by $h$ frames. $\odot$ denotes element-wise multiplication. Following this inpainting operation, we proceed to the denoising step with the updated $\hat{\boldsymbol{x}}_0$ using Eq. (6.2). We report the main results of our system with stride $h = 180$.

However, eliminating the need for future windows is insufficient for online inference with minimal latency since a new window of motion is generated only every $h$ frames, resulting in a latency of $(h-1) \times \delta t$, where $1/\delta t$ is the frame rate. We additionally report our system's results with $h = 10$, indicating a latency of just 0.17 seconds close to online requirements. Nonetheless, a smaller $h$ compromises motion quality, as it limits the use of future information. In general, $h$ can be a tunable parameter to trade off quality and latency.

## 6.3 Implementation details

**Image encoder.** We use CLIP [RKH+21] variation ViT-L/14 for our experiments and compute embeddings from the timestamp-synchronized 30 FPS camera; to get the 60 FPS image feature condition, we duplicate every frame one more time. We also tried other image encoders and found that CLIP features perform best for our task – please refer to Sec. 6.4.2 for experimental results.

**Pointcloud encoder.** The PC autoencoder consists of the encoder and decoder parts; the encoder consists of 4 convolution layers with $3 \times 3$ kernel, channel sizes 16,32,64,128 correspondingly, ReLU in between, with the average pooling in the end to produce one feature vector of size 128. Decoder is an inversion of that, consisting of 4 transposed convolution layers. It is trained on the volumes extracted using our train set's point clouds and head trajectories. We train with Adam [KB14] optimizer and learning rate of $10^{-3}$ for 10 epochs.

**Dataset and tranining setup.** To address the limitation of evaluating on synthetic or smaller real-world datasets, we train and evaluate our system on a large-scale, first-of-

its-kind real-device dataset Nymeria [MYH+24], coauthored by the author of this thesis. It is captured from Project Aria glasses [Pro24] paired with XSens [RLS09] IMU motion capture suit. The Project Aria glasses are set to record 30 FPS color video at $1408\times1408$ pixel resolution. Data captured from the glasses are further processed with its machine perception service (MPS) [ESG+23] to output the head transformation and point clouds. The XSens motion data is recorded onboard at 1KHz and processed with MVN Analyse Pro [Xse24] as 240Hz full-body motion, downsampled to 60Hz for our input. The body motion from XSens is synchronized with Aria data to high accuracy using a custom timecode device. The body motion is further calibrated to the Aria head transformation to reduce spatial drift.

The full dataset contains 1200 motion sequences totaling 300 hours of daily activities of 264 participants across 50 locations, from which we used 1040 due to spatial synchronization problems in some sequences. Participants are recruited to cover uniform demographics along the axes of gender, age, height, and weight. The locations include 47 AirBnbs, where 31 are multi-floor houses. Scene set also includes a cafeteria with an outdoor patio, a multistory office building, and a campus with a parking lot and multiple biking/hiking trails.

The dataset covers a wide range of daily activities. The highest occurrences are cooking (13.5%), searching objects (11.0%), free-form activity improvise (10.4%), and playing games (10.1%), whereas the lowest occurrences include working at a desk (1.6%), locomotion (2.2%), activities in the office (2.3%), and making a mess at home (2.3%). Outdoor activities consist approximately 15% of the data. For additional details of the dataset, we refer readers to the Nymeria paper [MYH+24].

We split the dataset for training/validation/testing as 806/10/224 sequences, corresponding to 202/3/56 hours. *The testing split does not contain any locations or subjects that appear in the training set* to ensure no data leakage. We also strive to maintain a similar distribution of activities between the training set and the test set.

We train our models on this dataset with a context window of 240 frames (4 sec) for 20 epochs, which takes 3.5 days with four NVIDIA A100 GPUs.

**System runtime.** The inference is done on a single Nvidia A100 GPU and achieves better than real-time throughput of $>70$ FPS with an online 0.17s-latency ($h=10$) model and $>1350$ FPS with high-latency (3s, $h=180$) model. Our implementation assumes that point cloud encodings and CLIP features are precomputed or computed in parallel on a separate device. The performance will be affected if all computations need to happen on the same device. However, we observed that even in this situation, we could achieve a throughput of $\sim 61$ FPS for our low-latency variant, keeping up with real-time speed:

CLIP embeddings take around 5 ms to compute per image (2.5 ms per motion frame since we are duplicating every frame), and point cloud encoder taking around 0.1 ms per motion frame. Note that the runtime performance is evaluated on a powerful GPU; optimizing this system to achieve real-time speeds on the head-mounted device itself is a promising future research direction. Additionally, our implementation assumes the access to all SLAM feature points in the approximately 15 minutes window of the whole motion sequence. In a true real-time setting, this simplification would require a warm-up phase in the same environment of similar time length.

## 6.4 Experiments

We conducted a set of experiments to support these claims:

- Our multi-modal conditioning improves motion quality.
- Our system achieves high reconstruction accuracy, motion diversity, and physical realism.
- Our online (low-latency) variant minimally degrades motion quality compared to high-latency inference.
- Our system improves results over the state-of-the-art baselines on a large-scale dataset.

**Baselines.** We benchmark our low- and high-latency systems against EgoEgo [LLW23] and AvatarPoser [JSQ+22], retraining both models on our dataset. For EgoEgo, we bypass its first stage, using Aria Glasses' SLAM for accurate head motion tracking, and test with its long-sequence inference code. For AvatarPoser, we only provide head motion, masking out wrist device input during training and testing. Unlike the Nymeria paper's short-segment evaluations [MYH+24], we test all methods with full motion sequences (each around 15 minutes) in an online, autoregressive setting, reflecting real-world use.

**Metrics.** An ideal conditional motion generation algorithm must balance reconstruction accuracy, motion diversity, and physical realism. For instance, when arms are visible to the HMD camera, generated motions should reflect that. When multiple motions are equally valid, *e.g.* sitting, squatting, or kneeling, predictions should cover all possibilities. Finally, any output motion should be visually realistic and within the distribution of physically plausible human movements. We choose metrics that evaluate a system's capability to balance these three goals.

- **Reconstruction:** we report joint position errors (Mean Per Joint Position Errors, MPJPE) for all methods. As we use the head frame from Aria as the body reference frame for all methods, we assume zero error on head positions or orientations. Instead, we report position errors of the wrist joints (Hand PE). In Sec. 6.4.2 we also report errors for the joints in the lower and upper half of the body (Low. PE and Up. PE).

- **Diversity:** following prior work [RLL+23, GZW+20], we report the diversity metric as the mean distance between two same-size randomly sampled subsets from predicted and ground-truth motions in the latent space. The latent space is formed by training an auto-encoder, following the protocol in Guo *et al*. 2020 [GZW+20].

- **Realism:** we report FID scores [GZW+20] measuring the distances in distributions between predicted and ground-truth motions in the same latent space used for the Diversity metric. We also report the physicality of motions, following the metric proposed in EDGE [TCL23], which correlates with foot sliding. Lastly, we report the mean floor penetration depth. Since the floor level varies across time and is non-trivial to estimate for outdoor and complex indoor environments (e.g. the "floor" height for lying in bed should sensibly be the bed height), we adopt a conservative proxy using the lowest joint position of the ground-truth motion across the neighboring 20 seconds.

All the metrics that have units of measure, namely positional errors (MPJPE, Hand PE, Low. PE, Up. PE) and Floor Penetration, are presented in cm. The down arrow $\downarrow$ means that lower value is always better for this metric, and the right arrow $\rightarrow$ means that the value closer to Ground-truth is better.

### 6.4.1 Main Results

We evaluated high- ($h = 180$) and low-latency ($h = 10$) variants of our system on the 56-hour (224 sequences) test split, averaging 15 minutes per sequence. These test sequences are **not** cut into short segments to fit the temporal horizon $T$ of the model – all models are tasked to generate the entire sequence coherently, which is closer to practical application setup. Unlike EgoEgo, where statistics are reported using the best among 200 repetitions, we report the mean and standard deviation of all repetitions. As our test set is very large (e.g. the AMASS [MGT+19] testing subset used in AvatarPoser contains just two hours of motion), we only run eight repetitions for each of the 224 sequences.

| | MPJPE ↓ | Hand PE ↓ | FID ↓ | Diversity → | Physicality → | Floor Pen. ↓ |
|---|---|---|---|---|---|---|
| Ground-truth | 0 | 0 | 0 | 16.13 | 0.56 | 0 |
| EgoEgo | $16.61^{\pm 1.49}$ | $34.64^{\pm 1.64}$ | $35.69^{\pm 0.54}$ | $20.15^{\pm 0.21}$ | $3.68^{\pm 0.74}$ | $2.43^{\pm 1.54}$ |
| AvatarPoser (Head) | 10.64 | 21.51 | 27.61 | 12.99 | 1.69 | 4.21 |
| Ours ($h = 180$) | $\mathbf{8.36}^{\pm 0.08}$ | $\mathbf{16.64}^{\pm 0.21}$ | $\mathbf{2.16}^{\pm 0.02}$ | $\mathbf{15.74}^{\pm 0.29}$ | $\mathbf{1.03}^{\pm 0.01}$ | $\mathbf{1.03}^{\pm 0.06}$ |
| Ours ($h = 10$) | $9.19^{\pm 0.05}$ | $17.67^{\pm 0.06}$ | $5.00^{\pm 0.02}$ | $15.23^{\pm 0.02}$ | $1.30^{\pm 0.10}$ | $1.19^{\pm 0.04}$ |

Table 6.1: Quantitative results comparing our system with EgoEgo and AvatarPoser.

| | MPJPE ↓ | Hand PE ↓ | FID ↓ | Diversity → | Physicality → | Floor Pen. ↓ |
|---|---|---|---|---|---|---|
| Ground-truth | 0 | 0 | 0 | 16.13 | 0.56 | 0 |
| Ours w/ DINOv2 | $8.72^{\pm 0.07}$ | $17.24^{\pm 0.18}$ | $2.45^{\pm 0.02}$ | $15.38^{\pm 0.19}$ | $\mathbf{0.91}^{\pm 0.00}$ | $1.42^{\pm 0.07}$ |
| Ours w/ VC-1 | $8.54^{\pm 0.11}$ | $\mathbf{16.64}^{\pm 0.22}$ | $4.34^{\pm 0.06}$ | $15.00^{\pm 0.42}$ | $0.92^{\pm 0.01}$ | $1.26^{\pm 0.10}$ |
| Ours w/ CLIP (current) | $\mathbf{8.36}^{\pm 0.08}$ | $\mathbf{16.64}^{\pm 0.21}$ | $\mathbf{2.16}^{\pm 0.02}$ | $\mathbf{15.74}^{\pm 0.29}$ | $1.03^{\pm 0.01}$ | $\mathbf{1.03}^{\pm 0.06}$ |

Table 6.2: Comparison between different image feature encoders. MPJPE, Hand PE and Floor penetration are in cm.

**Quantitative Results.** The main quantitative results are summarized in Table 6.1. Our system achieved superior performance across all three metric axes of reconstruction, diversity, and realism. The online variant of our system degrades performance slightly, as expected, given its inaccessibility of future sensor information.

Our adapted version of AvatarPoser (referred to as AvatarPoser (Head) in Table 6.1) performs well, but its frame-by-frame prediction can lack temporal coherence, reducing realism. As a regression model, it captures only the average trend in training data, leading to lower diversity scores. EgoEgo also generates visually reasonable motions but has two key issues. Despite its diffusion-based long motion inference, discontinuities in long motions affect realism metrics. Additionally, EgoEgo tends to produce overly dynamic arm movements, similar to how some image diffusion models create stylized rather than naturalistic outputs. This leads to higher Hand Position Errors and contributes to increased MPJPE and Diversity scores compared to ground truth. Unlike our multi-modal approach, the baseline methods do not incorporate environmental context, which reduces their performance compared to our system (Fig. 6.5). While all the metrics are measured as mean across all runs, we additionally report MPJPE of the best-case run: *8.246 cm* for $\text{HMD}^2$ and *14.678 cm* for EgoEgo (AvatarPoser is deterministic, so stays the same). Compared to Table 6.1, errors for EgoEgo are noticeably lower but are still behind our method.

In summary, our system uniquely balances the accuracy of motion reconstruction and fidelity and diversity of motion generation, surpassing baseline methods. The online

| | MPJPE ↓ | Hand PE ↓ | FID ↓ | Diversity → | Physicality → | Floor Pen. ↓ |
|---|---|---|---|---|---|---|
| Ground-truth | 0 | 0 | 0 | 16.13 | 0.56 | 0 |
| Ours, w/o PC, w/o CLIP | $9.28^{\pm 0.23}$ | $19.47^{\pm 0.36}$ | $6.75^{\pm 0.08}$ | $14.44^{\pm 0.30}$ | $0.90^{\pm 0.01}$ | $3.29^{\pm 0.31}$ |
| Ours, w/ PC, w/o CLIP | $8.97^{\pm 0.10}$ | $20.38^{\pm 0.28}$ | $3.68^{\pm 0.03}$ | $15.29^{\pm 0.42}$ | $\mathbf{0.86}^{\pm 0.00}$ | $\mathbf{0.99}^{\pm 0.07}$ |
| Ours, w/o PC, w/ CLIP | $8.57^{\pm 0.11}$ | $\mathbf{16.32}^{\pm 0.22}$ | $6.17^{\pm 0.02}$ | $14.79^{\pm 0.22}$ | $1.01^{\pm 0.01}$ | $2.15^{\pm 0.15}$ |
| Ours, w/ PC, w/ CLIP | $\mathbf{8.36}^{\pm 0.08}$ | $16.64^{\pm 0.21}$ | $\mathbf{2.16}^{\pm 0.02}$ | $\mathbf{15.74}^{\pm 0.29}$ | $1.03^{\pm 0.01}$ | $1.03^{\pm 0.06}$ |

Table 6.3: **Ablation study.** HMD$^2$ leverages both point cloud (PC) and egocentric video information (CLIP) to reduce per-joint error while keeping the realism and physical plausibility of the motions.

variant of our system achieves 0.17-second latency with only a slight degradation in terms of performance, though the gap leaves room for future research and improvement.

**Qualitative Examples.** Fig. 6.5 visually compares all methods on two motion subsequences from the test set. *Sequence 1* shows a complex transition from kneeling to sitting. Regression models like AvatarPoser struggle in under-constrained scenarios, either abruptly switching between poses or averaging them into unnatural ones (e.g., a floating avatar in the last frame). EgoEgo, as a generative model, produces plausible motions but lacks the context to match the ground truth given only head motion. *Sequence 2* demonstrates another important advantage of our model – making use of the semantic features from color images. In this ground truth motion, the hands are raised and visible in the camera alternately. We successfully reproduce similar arm movements by conditioning on the CLIP embeddings while both baselines have the arms down.

The generative nature of our model also allows us to produce diverse motions in case of ambiguities. Fig. 6.6 and Fig. 6.7 show several examples. In sequences A and B of Fig. 6.6, our model generates various plausible states when hands are not visible, such as different poses for the non-visible left hand (seq. A). Sequences C and D show cases with equally possible leg positions, like kneeling vs. squatting (seq. C). Fig. 6.7 shows 4 random motion samples given the same input for two sequences (1st sequence indoor, 2nd sequence outdoor). A few observations worth highlighting:

- EgoEgo is capable of generating diverse predictions, sometimes more diverse than Ours. However, EgoEgo generations tend to be of lower quality - possibly due to model architecture not being as scalable to a massive dataset as Ours and autoregressive long sequence inference not working as well;

| | MPJPE ↓ | Hand PE ↓ | Low. PE ↓ | Up. PE ↓ | Floor Pen. ↓ |
|---|---|---|---|---|---|
| EgoEgo | $16.61^{\pm 1.49}$ | $34.64^{\pm 1.64}$ | $26.58^{\pm 3.57}$ | $11.31^{\pm 0.54}$ | $2.43^{\pm 1.54}$ |
| AvatarPoser (Head) | 10.64 | 21.51 | 17.70 | 6.90 | 2.94 |
| AvatarPoser (Head & Hands) | **7.74** | **6.29** | 16.10 | **3.11** | 4.63 |
| Ours, w/o PC, w/o CLIP | $9.28^{\pm 0.23}$ | $19.47^{\pm 0.36}$ | $15.04^{\pm 0.53}$ | $6.21^{\pm 0.11}$ | $3.29^{\pm 0.31}$ |
| Ours, w/ PC, w/o CLIP | $8.97^{\pm 0.10}$ | $20.38^{\pm 0.28}$ | $\underline{13.59}^{\pm 0.21}$ | $6.53^{\pm 0.07}$ | $\mathbf{0.99}^{\pm 0.07}$ |
| Ours, w/o PC, w/ CLIP | $8.57^{\pm 0.11}$ | $\underline{16.32}^{\pm 0.22}$ | $14.02^{\pm 0.25}$ | $\underline{5.64}^{\pm 0.06}$ | $2.15^{\pm 0.15}$ |
| Ours, w/ PC, w/ CLIP | $\underline{8.36}^{\pm 0.08}$ | $16.64^{\pm 0.21}$ | $\mathbf{13.23}^{\pm 0.16}$ | $5.75^{\pm 0.06}$ | $\underline{1.03}^{\pm 0.06}$ |

Table 6.4: Lower and upper body error depending on the input variations. We are beating a 3-point input baseline on a lower body error and achieve close performance on average. All the metrics are in cm.

| | MPJPE ↓ | Hand PE ↓ | FID ↓ | Diversity → | Physicality → | Floor Pen. ↓ |
|---|---|---|---|---|---|---|
| Ground-truth | 0 | 0 | 0 | 16.95 | 0.04 | 0 |
| $h = 230$ | $9.53^{\pm 0.01}$ | $16.15^{\pm 0.04}$ | $\mathbf{13.44}^{\pm 0.01}$ | $15.28^{\pm 0.01}$ | $0.32^{\pm 0.00}$ | $1.47^{\pm 0.02}$ |
| $h = 220$ | $9.49^{\pm 0.02}$ | $16.07^{\pm 0.06}$ | $13.61^{\pm 0.01}$ | $15.30^{\pm 0.01}$ | $0.25^{\pm 0.00}$ | $1.46^{\pm 0.01}$ |
| $h = 200$ | $9.44^{\pm 0.01}$ | $\mathbf{16.03}^{\pm 0.04}$ | $13.74^{\pm 0.01}$ | $15.32^{\pm 0.01}$ | $0.23^{\pm 0.00}$ | $1.45^{\pm 0.02}$ |
| $h = 180$ (Ours) | $\mathbf{9.42}^{\pm 0.02}$ | $16.05^{\pm 0.02}$ | $13.76^{\pm 0.01}$ | $15.43^{\pm 0.01}$ | $\mathbf{0.22}^{\pm 0.00}$ | $1.44^{\pm 0.01}$ |
| $h = 120$ | $9.43^{\pm 0.03}$ | $16.05^{\pm 0.05}$ | $14.02^{\pm 0.01}$ | $15.22^{\pm 0.01}$ | $0.26^{\pm 0.00}$ | $1.43^{\pm 0.01}$ |
| $h = 60$ | $9.49^{\pm 0.06}$ | $16.19^{\pm 0.03}$ | $14.23^{\pm 0.01}$ | $15.20^{\pm 0.01}$ | $0.30^{\pm 0.00}$ | $1.33^{\pm 0.03}$ |
| $h = 30$ | $9.61^{\pm 0.04}$ | $16.42^{\pm 0.07}$ | $14.39^{\pm 0.03}$ | $15.57^{\pm 0.03}$ | $0.40^{\pm 0.00}$ | $1.26^{\pm 0.03}$ |
| $h = 20$ | $9.75^{\pm 0.10}$ | $16.51^{\pm 0.08}$ | $16.46^{\pm 0.04}$ | $15.36^{\pm 0.04}$ | $0.45^{\pm 0.00}$ | $\mathbf{1.18}^{\pm 0.05}$ |
| $h = 10$ (Ours low-lat.) | $10.19^{\pm 0.12}$ | $17.13^{\pm 0.14}$ | $17.00^{\pm 0.10}$ | $15.66^{\pm 0.10}$ | $0.73^{\pm 0.03}$ | $1.41^{\pm 0.14}$ |
| $h = 5$ | $13.13^{\pm 0.46}$ | $21.28^{\pm 0.45}$ | $20.36^{\pm 0.33}$ | $\mathbf{16.71}^{\pm 0.33}$ | $0.94^{\pm 0.02}$ | $1.84^{\pm 0.43}$ |
| $h = 3$ | $21.10^{\pm 1.08}$ | $29.80^{\pm 1.15}$ | $72.63^{\pm 0.82}$ | $20.35^{\pm 0.82}$ | $1.29^{\pm 0.12}$ | $4.49^{\pm 0.51}$ |
| $h = 1$ | $28.96^{\pm 1.68}$ | $38.13^{\pm 1.54}$ | $129.94^{\pm 1.37}$ | $22.74^{\pm 1.37}$ | $2.22^{\pm 0.17}$ | $3.75^{\pm 0.72}$ |

Table 6.5: Ablation study on the latency ($h$) parameter. Test is performed on a subset (9%) of the current test split. MPJPE, Hand PE and Floor penetration are in cm.

- EgoEgo samples often do not satisfy floor height constraints (1st seq. 3rd frame; 2nd seq. 1st frame), and cannot utilize image observation when certain body parts are visible (1st seq., see the right arm in 1st frame and left arm in 2nd frame);

- Samples from our method are "conditionally diverse". This is unseen in previous works. For example, when the egocentric camera sees only one arm, Ours will generate samples with this arm doing the motion seen (not perfectly accurate partially due to CLIP) and generate motions for the unseen arm and legs with diversity (see arms in 1st&2nd frames on the 1st sequence, see legs in all frames on the second sequence).

| | MPJPE ↓ | Hand PE ↓ | Low. PE ↓ | Up. PE ↓ | Floor Pen. ↓ |
|---|---|---|---|---|---|
| EgoEgo | $30.91^{\pm 4.82}$ | $60.81^{\pm 2.98}$ | $58.63^{\pm 12.17}$ | $19.26^{\pm 1.16}$ | $10.33^{\pm 5.90}$ |
| AvatarPoser (Head) | 22.09 | 43.19 | 44.18 | 13.01 | 18.96 |
| AvatarPoser (Head & Hands) | 16.48 | **11.23** | 37.91 | **5.63** | 18.15 |
| Ours, w/o PC, w/o CLIP | $18.31^{\pm 0.89}$ | $40.15^{\pm 1.17}$ | $34.35^{\pm 2.20}$ | $11.75^{\pm 0.37}$ | $12.91^{\pm 1.75}$ |
| Ours, w/ PC, w/o CLIP | $16.65^{\pm 0.44}$ | $41.68^{\pm 1.05}$ | $28.72^{\pm 1.02}$ | $12.29^{\pm 0.30}$ | $\mathbf{3.97}^{\pm 0.32}$ |
| Ours, w/o PC, w/ CLIP | $16.30^{\pm 0.55}$ | $34.25^{\pm 0.90}$ | $29.98^{\pm 1.35}$ | $10.58^{\pm 0.26}$ | $8.28^{\pm 0.78}$ |
| Ours, w/ PC, w/ CLIP | $\mathbf{15.49}^{\pm 0.38}$ | $34.86^{\pm 0.92}$ | $\mathbf{27.35}^{\pm 0.81}$ | $10.80^{\pm 0.26}$ | $4.22^{\pm 0.28}$ |

Table 6.6: Lower and upper body error study on top 5% errors (mean of 95% percentiles across all sequences). Here, we are beating 3-point error baseline on mean per-joint positional error. All the metrics are in cm.

| | MPJPE ↓ | Hand PE ↓ | FID ↓ | Diversity → | Physicality → | Floor Pen. ↓ |
|---|---|---|---|---|---|---|
| Ground-truth | 0 | 0 | 0 | 16.95 | 0.04 | 0 |
| 2 steps | $9.54^{\pm 0.01}$ | $15.94^{\pm 0.02}$ | $15.04^{\pm 0.00}$ | $15.45^{\pm 0.00}$ | $0.50^{\pm 0.00}$ | $1.87^{\pm 0.02}$ |
| 3 steps | $9.27^{\pm 0.01}$ | $\mathbf{15.52}^{\pm 0.03}$ | $15.28^{\pm 0.01}$ | $14.85^{\pm 0.01}$ | $0.32^{\pm 0.00}$ | $1.64^{\pm 0.01}$ |
| 5 steps | $\mathbf{9.26}^{\pm 0.01}$ | $15.57^{\pm 0.03}$ | $14.94^{\pm 0.01}$ | $14.97^{\pm 0.01}$ | $0.25^{\pm 0.00}$ | $1.54^{\pm 0.02}$ |
| 10 steps | $9.34^{\pm 0.02}$ | $15.81^{\pm 0.03}$ | $14.25^{\pm 0.01}$ | $15.50^{\pm 0.01}$ | $0.24^{\pm 0.00}$ | $1.47^{\pm 0.01}$ |
| 20 steps (Ours) | $9.42^{\pm 0.02}$ | $16.05^{\pm 0.02}$ | $13.76^{\pm 0.01}$ | $15.43^{\pm 0.01}$ | $\mathbf{0.22}^{\pm 0.00}$ | $1.44^{\pm 0.01}$ |
| 40 steps | $9.52^{\pm 0.02}$ | $16.21^{\pm 0.02}$ | $13.40^{\pm 0.01}$ | $15.71^{\pm 0.01}$ | $0.22^{\pm 0.00}$ | $1.43^{\pm 0.02}$ |
| 80 steps | $9.60^{\pm 0.03}$ | $16.38^{\pm 0.02}$ | $\mathbf{13.11}^{\pm 0.01}$ | $\mathbf{15.77}^{\pm 0.01}$ | $0.23^{\pm 0.00}$ | $\mathbf{1.41}^{\pm 0.01}$ |

Table 6.7: Ablation study on the amount of steps in reverse diffusion process. Test is performed on a subset (10%) of the current test split.

### 6.4.2 Additional Analysis

**Comparison between different images feature encoders.** To explain our choice of CLIP [RKH+21] feature as a feature encoder, we additionally trained two versions of our method with image features produced by DINOv2 [ODM+23] and VC-1 [MYA+24] feature encoders. For VC-1, we chose the best performing ViT-L model, with embedding size of 1024 and input size of $250 \times 250$ (cropped to $224 \times 224$ during preprocessing); for DINOv2, we chose second to largest model ViT-L/14, providing it with the input of the same size (padded to $252 \times 252$) and taking the class token of the output (size 1024), which corresponds to the global image description as it gathers the information from all the image patches. The comparison is presented in Tab. 6.2. We found that, while methods VC-1 and DINOv2 have close generation precision and a slight advantage in Physicality (correlated to foot sliding), the model with CLIP features shows the best results on most metrics, proving our choice of the image feature encoder.

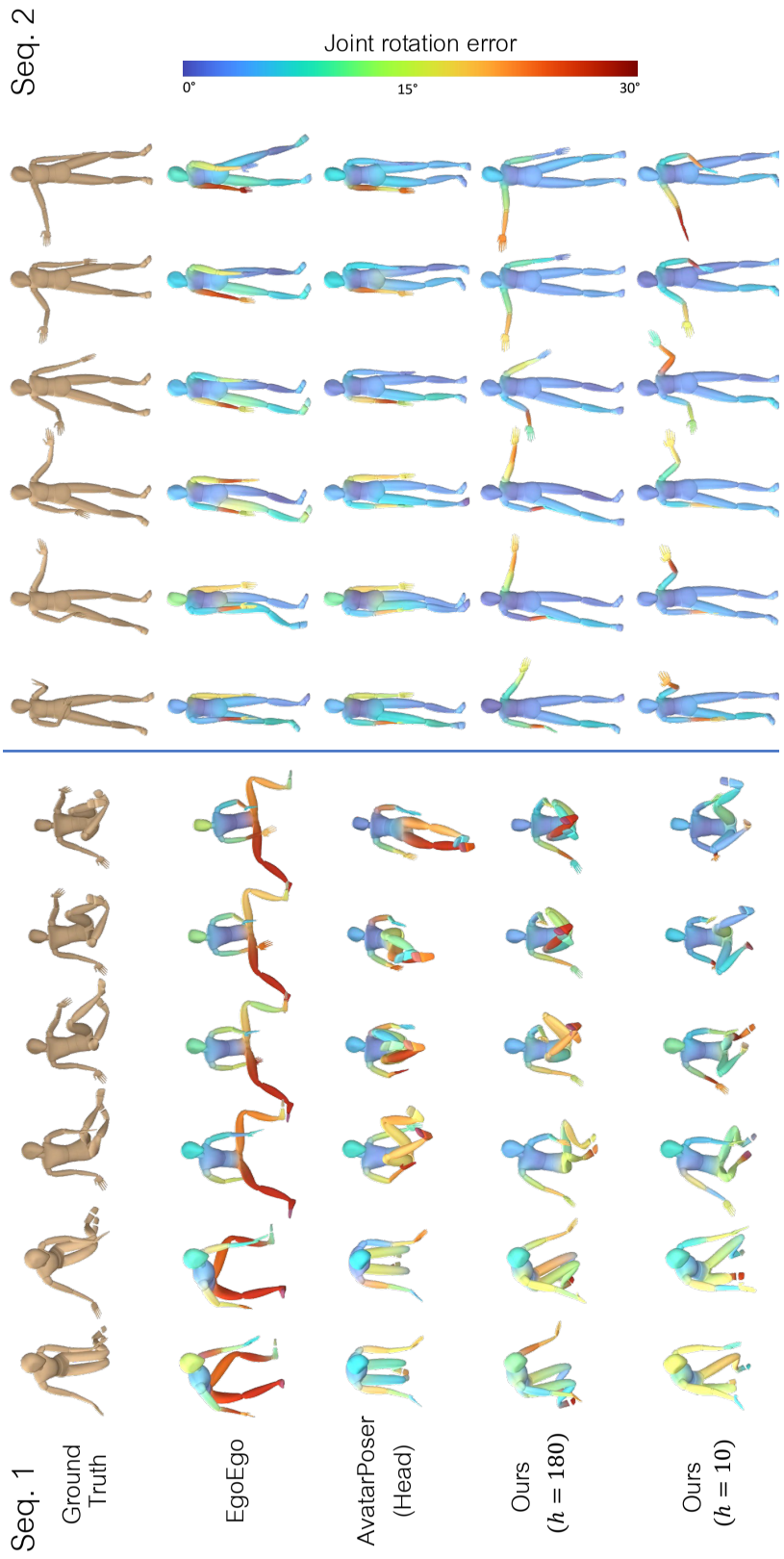



Figure 6.5: Qualitative comparison between HMD$^2$ (Ours) and baseline methods.

**Ablations on input variations.** We ablated our system by removing the point cloud encoder branch (w/o PC) and/or the raw egocentric video branch (w/o CLIP). The results are summarized in Table 6.3, demonstrating the importance of multi-modal scene and motion conditions in our system.

Even without point cloud and CLIP embeddings, our system generates temporally coherent and realistic full-body motions, capturing diverse motion distributions. However, ambiguity arises with head movement alone, such as distinguishing between standing and sitting. Without environmental context, the system might randomly generate or switch between these actions, affecting realism metrics (FID & Floor Penetration depth). Table 6.3 shows that point cloud embeddings help align motions with ground truth and reduce environment interpenetration, improving realism. The image encoder also enhances reconstruction accuracy by using semantic clues, particularly when hands are visible. This reduces MPJPE by encouraging specific poses, however it also mildly affects the realism of motion, hence Physicality metric slightly degrades. Fig. 6.8 illustrates that PC embeddings enable correct sitting motion detection, while image embeddings improve hand motion accuracy. Together, they produce more accurate and realistic results.

**Additional quantitative results.** In Table 6.4, we present additional metrics, splitting per-joint average error into average error across upper (Up. PE) and lower (Low. PE) body regions. The upper region is defined as all the joints that are higher than the pelvis for the subject standing in a T-pose, namely the spine, shoulders, arms, hands, neck, and head. The lower body region is defined as the rest of the joints, excluding the root joint (hips, legs, feet). From these metrics, we can directly observe the effect of adding pointcloud and image encoders to our data. When the PC encoder is added, the lower body error is reduced significantly, and the upper body gets slightly worse (most likely due to noisy points near the upper body region). This suggests that pointcloud helps to disambiguate the lower body by providing landscape information (floor level, nearby objects, etc.). On the other hand, when CLIP image encoding is added, we notice a major reduction in the upper body error, suggesting that image features help the method better understand interactions and localize hands. At the same time, lower body error also decreases - most likely, the error is reduced when parts of the lower body are visible on camera. HMD$^2$, denoted as "Ours, w/ PC, w/ CLIP" in the table, combines both strengths of the methods above and achieves the lowest mean per-joint error.

At the same table, we also present a study of another, much more challenging baseline – a 3-point input method. For that, we chose the original implementation AvatarPoser [JSQ+22], which takes not only the head position and orientation as an input but also the positions and orientations of the hands. With more input information, this baseline

| | MPJPE ↓ | Hand PE ↓ | Low. PE ↓ | Up. PE ↓ | Floor Pen. ↓ |
|---|---|---|---|---|---|
| EgoEgo | $12.06^{\pm 0.33}$ | $31.31^{\pm 1.13}$ | $17.24^{\pm 0.75}$ | $9.40^{\pm 0.30}$ | $\mathbf{0.01}^{\pm 0.00}$ |
| AvatarPoser (Head) | 7.39 | 14.81 | 12.58 | 4.64 | 0.11 |
| Ours ($h = 180$) | $\mathbf{5.75}^{\pm 0.03}$ | $\mathbf{11.98}^{\pm 0.13}$ | $\mathbf{8.84}^{\pm 0.07}$ | $\mathbf{4.06}^{\pm 0.03}$ | $0.02^{\pm 0.00}$ |
| Ours ($h = 10$) | $6.19^{\pm 0.04}$ | $12.16^{\pm 0.07}$ | $9.97^{\pm 0.10}$ | $4.13^{\pm 0.01}$ | $0.02^{\pm 0.00}$ |

Table 6.8: Results for the scenario with the best HMD$^2$ performance. Scenario is consisting of the multi-terrain outdoor walking (hiking up- and downhill), mostly sightseeing. All the metrics are in cm.

| | MPJPE ↓ | Hand PE ↓ | Low. PE ↓ | Up. PE ↓ | Floor Pen. ↓ |
|---|---|---|---|---|---|
| EgoEgo | $12.29^{\pm 0.25}$ | $32.32^{\pm 0.50}$ | $16.40^{\pm 0.64}$ | $10.16^{\pm 0.16}$ | $\mathbf{0.31}^{\pm 0.14}$ |
| AvatarPoser (Head) | 8.39 | 20.94 | 11.44 | 6.78 | 0.80 |
| Ours ($h = 180$) | $\mathbf{6.53}^{\pm 0.06}$ | $\mathbf{15.66}^{\pm 0.17}$ | $\mathbf{8.86}^{\pm 0.10}$ | $\mathbf{5.29}^{\pm 0.05}$ | $0.42^{\pm 0.05}$ |
| Ours ($h = 10$) | $7.32^{\pm 0.05}$ | $17.30^{\pm 0.17}$ | $10.05^{\pm 0.10}$ | $5.87^{\pm 0.04}$ | $0.45^{\pm 0.02}$ |

Table 6.9: Results for the scenario with the median across all 20 scenarios HMD$^2$ performance. Scenario is consisting of flat-ground indoor multi-room interactions with the objects in the house (grabbing clothes, throwing pillows, opening doors), mostly upright standing with occasional bending (to reach for the next object). All the metrics are in cm.

| | MPJPE ↓ | Hand PE ↓ | Low. PE ↓ | Up. PE ↓ | Floor Pen. ↓ |
|---|---|---|---|---|---|
| EgoEgo | $28.67^{\pm 1.97}$ | $42.85^{\pm 1.46}$ | $52.11^{\pm 4.52}$ | $15.75^{\pm 0.64}$ | $12.76^{\pm 3.55}$ |
| AvatarPoser (Head) | 23.30 | 31.11 | 45.01 | 11.32 | 21.79 |
| Ours ($h = 180$) | $\mathbf{17.21}^{\pm 0.20}$ | $\mathbf{24.39}^{\pm 0.36}$ | $\mathbf{31.27}^{\pm 0.50}$ | $\mathbf{9.45}^{\pm 0.13}$ | $\mathbf{3.32}^{\pm 0.24}$ |
| Ours ($h = 10$) | $18.74^{\pm 0.65}$ | $26.28^{\pm 0.50}$ | $33.37^{\pm 1.41}$ | $10.55^{\pm 0.27}$ | $5.01^{\pm 0.39}$ |

Table 6.10: Results for the scenario with the worst HMD$^2$ performance. Scenario is consisting of challenging body stretching and yoga motions, mostly on done the floor, recorded indoors. All the metrics are in cm.

achieves better performance on average. However, we highlight that even with additional motion input, it is worse than Ours at generating lower body motion, as Lower body PE is higher. It is important to note that HMD$^2$ achieves *best performance* on the most challenging frames of the sequences even when compared to a 3-point input baseline, as shown in the top 5% error study in Tab. 6.6.

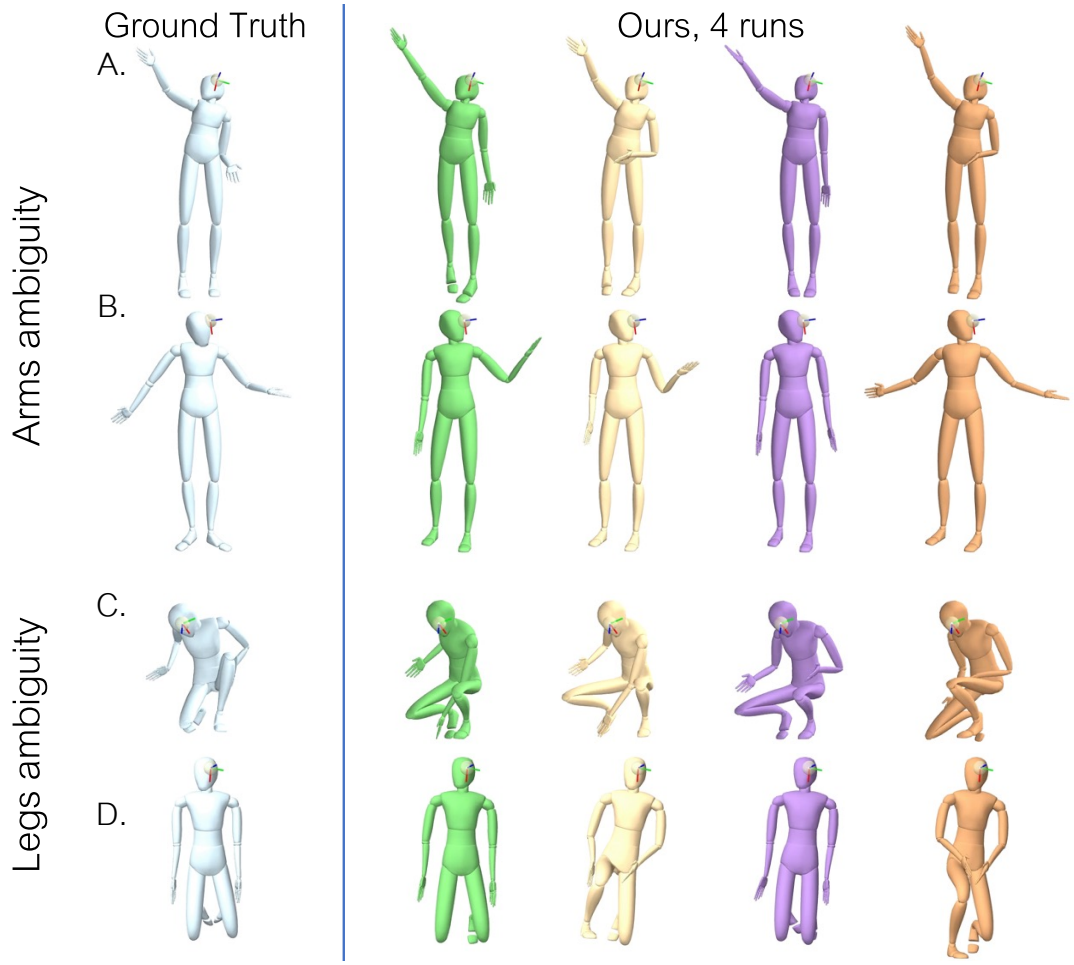



Figure 6.6: Our system can predict diverse outcomes from identical input (head pose marked as a sphere with coordinate system).

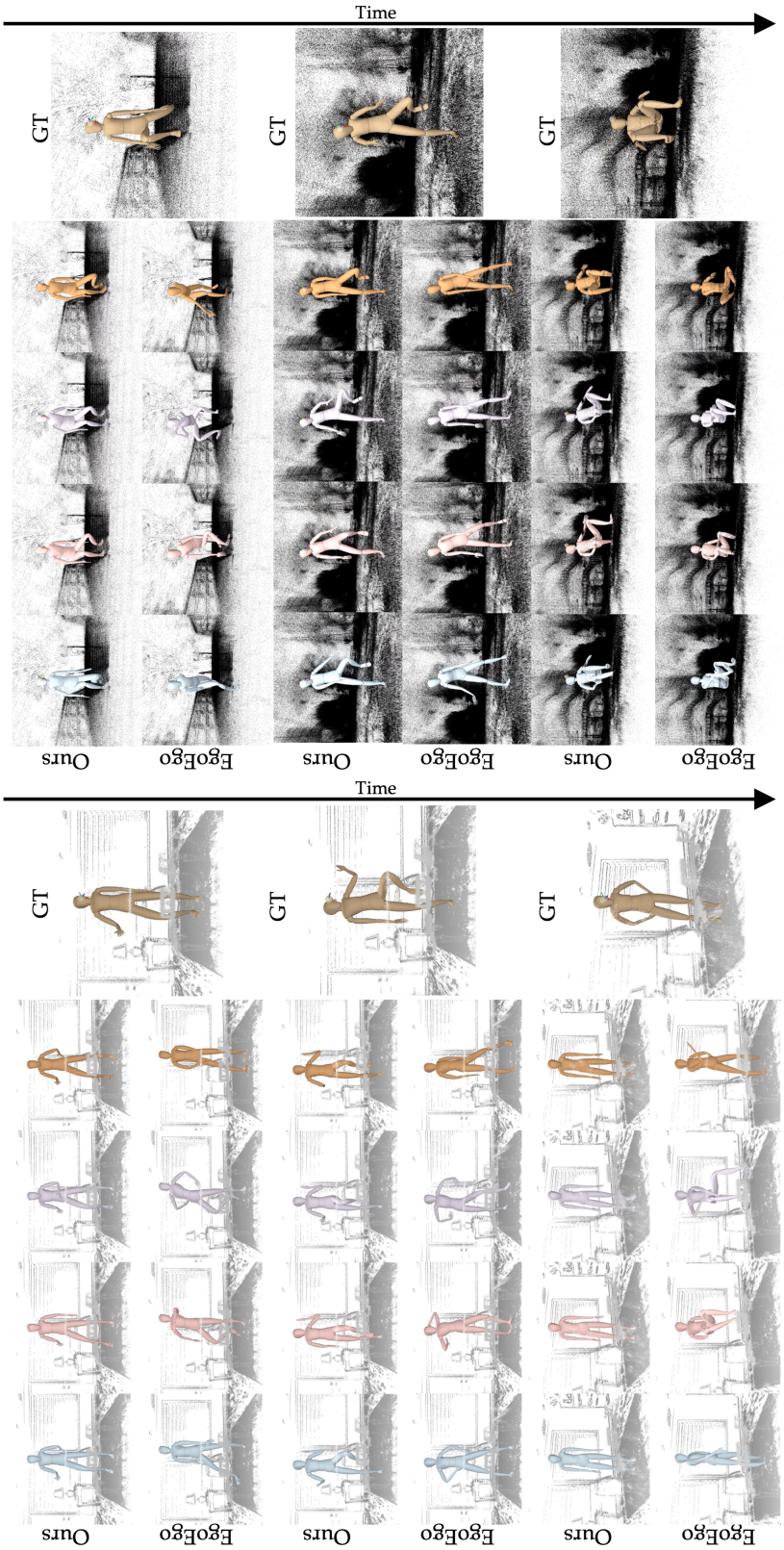



Figure 6.7: Range of possible results given the same input for HMD$^2$ and EgoEgo. Colors denote different runs, sequence frame time is increasing from top to bottom.

**Error Distribution.** As we evaluate on a large scale dataset of realistic daily activities, the metric statistics could be skewed and dominated by mundane actions such as sitting or standing still, or walking from A to B. The more interesting and challenging scenarios that highlight core issues may fall into a long-tail distribution and be obscured by the mean error. To this end, we also report the top 5% errors in Table 6.6, which is more representative of improvements we expect from our approach.

Our top error selection strategy can be explained as follows: as shown in Fig. 6.9, the average error on the sequence greatly depends on the activity performed in that sequence. If we were to sort all the per-frame joint errors and select the top 5% (95% percentile) among them, we would only select the frames from several worse-performing sequences. To avoid such behavior, we compute the 95% error percentile within each sequence separately and average those results across all sequences.

**Ablation study on $h$ parameter values and diffusion steps.** In Tab. 6.5, we show how the error metrics change depending on the latency ($h$) parameter. Because experiments with $h = 1$, 3, and 5 take a long time to process on our large test split, we performed this ablation on a 9% (20 out of 224 sequences) subset of test data. To keep the subset informative and maintain the diversity of activities, we picked one random sequence from each activity scenario. The results in the table demonstrate that the top performance in terms of MPJPE is achieved at $h = 180$, which we chose as our default value. While it is not the best on all the metrics, the difference is not as significant. Our low-latency method ($h = 10$) demonstrates some performance drop, but not as big compared to the next value $h = 5$, keeping a balance between the quality and the output lag.

We also measured metrics change w.r.t. the amount of diffusion steps we taking during inference. Tab. 6.7 shows that FID score increases with the amount of steps – visually, this corresponds to less jittery and more realistic motion. However, the precision of the motion, measured by MPJPE metric, peaks at 5 steps for full body and 3 steps for hands. Therefore, our choice of 20 steps is a balance between motion precision and realism.

**Variation of an error depending on the activity.** Our test dataset consists of diverse activities, and each sequence is dedicated to a certain type of activity according to the assigned scenario. In total, there are 20 scenarios, with indoor and outdoor activities featuring walking, sitting, laying, exercising, interacting with household objects, playing sports games, and more. If we group the sequences and measure the MPJPE in each group (Fig. 6.9), we can observe that the error is not distributed evenly – while for most scenarios the error does not exceed 8cm, there is a chunk of challenging scenarios that have an error almost twice as high. To understand the reasons behind this, we selected and studied different metrics for the scenario, including the best, the worst, and median

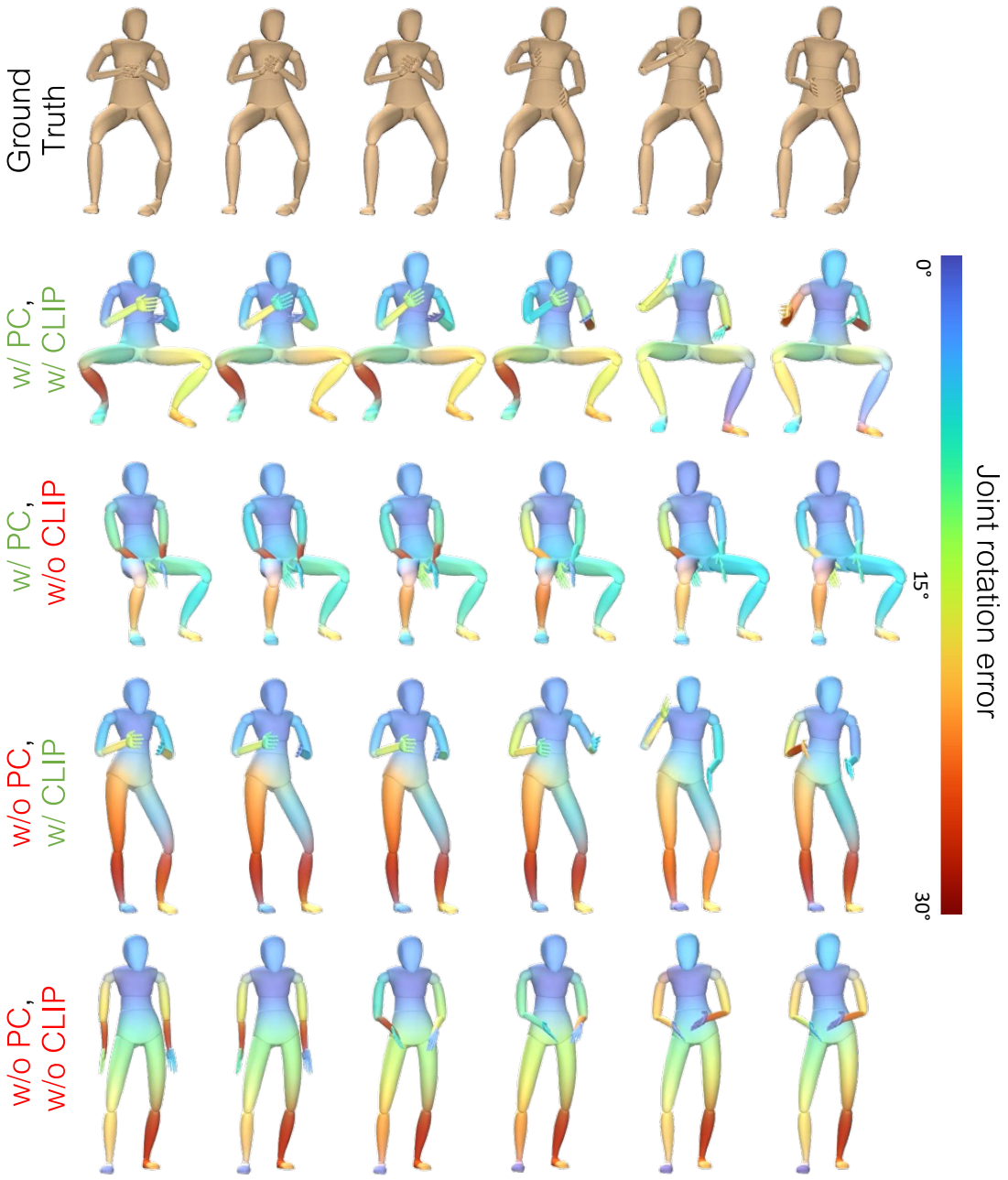



Figure 6.8: Example motion when ablating the point cloud (PC) or video (CLIP) branches.

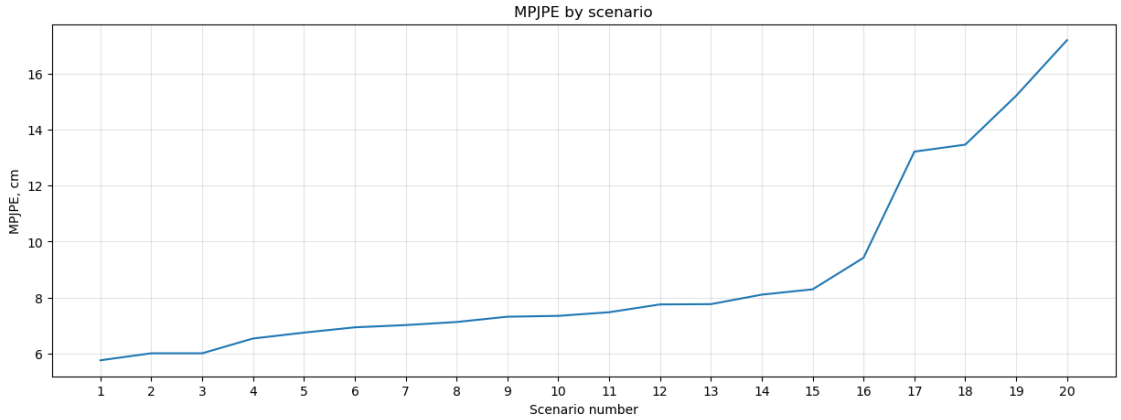



Figure 6.9: MPJPE depending on the action scenario (sorted in increasing order).

MPJPE. Results are presented in tables 6.8, 6.9, 6.10.

The best-performing scenario (Tab. 6.8) consists of multi-terrain outdoor walking (hiking up and downhill) but does not feature any interactions. Small lower body error demonstrates that multi-level motion is, in general, not a significant challenge for our method – in contrast to AvatarPoser, whose lower body error is higher on this scenario than on the mostly flat scenario from Tab. 6.9.

The scenario with the median method performance (Tab. 6.9) consists of mostly flat-ground indoor multi-room interactions with the objects in the house (grabbing clothes, throwing pillows, opening doors). The subject often stays in the standing position, occasionally bending to reach some objects. As interactions with the objects appear more often here, we notice higher hand positional errors for our method. This can be explained by the inability of the CLIP-encoded image features to localize the hands precisely during the interactions. Occasional bending can also be misinterpreted for a different motion sometimes, which explains higher floor penetration error.

The worst performing scenario (Tab. 6.10) consists mainly of yoga and body stretching motions, which proved to be the most challenging for all the methods. While the upper body error is higher than usual, the error is primarily increasing due to very high lower body error. This is caused by a high position uncertainty: most of the time, lower body parts are not observed by the camera, and the floor estimation from a SLAM point cloud might be noisy. Future work on improving the performance in such scenarios can benefit from enhancing the reconstructed SLAM pointcloud quality to provide reliable terrain information, including more of these challenging motions in the dataset and using cameras with a higher field of view, like fisheye cameras, to increase the body parts visibility.

## 6.5 Conclusions

We presented a diffusion-based framework, HMD$^2$, for online motion generation from a single head-mounted device. By leveraging camera streams for learning-based image embeddings and combining them with SLAM-derived head trajectories and semi-dense point clouds, our framework can produce diverse and natural motions that align with environmental contexts. We assessed our system across various environments and an extensive range of daily activities. Compared to existing state-of-the-art methods, HMD$^2$ significantly enhances motion quality in terms of accuracy, diversity, and realism.

Our insight into leveraging egocentric image features and the capabilities of modern SLAM systems opens up many new opportunities. For instance, in the future, we can incorporate more comprehensive contextual information from recent advancements in image understanding, including depth estimation from monocular videos, panoptic segmentation, or scene reconstruction through neural radiance fields or 3D Gaussian Splats. Additionally, we envision leveraging video embeddings over extended context windows, potentially from visual language models (VLMs) [AAA+23], to refine context conditions further.

Currently, the performance of our system is still limited by available context information. For example, the CLIP embeddings cannot provide precise spatial information, so they fall short of constraining the precise pose of the hands even when they are visible. The noisy and sparse point clouds are less suitable compared to dense depth maps for accurate environment contact information; the errors from the SLAM reconstruction can also propagate to our system. On the other hand, incorporating denser input streams poses a challenge in runtime performance.

Features that contain more precise positional information than CLIP may improve performance: one potential direction for future work is to additionally condition the method on the results of the hand-tracking algorithm. However, even without explicit positional information, CLIP-encoded images improve upper body tracking. The effect on the lower body is less apparent. This, of course, can be explained by the fact that the lower body is much less visible from the camera, especially since we use a camera with the standard FOV looking outwards. Additional information from the downward-looking wide-angle cameras can improve the performance, as shown in other works [WLX+23].

Even with the point cloud context provided, our method can sometimes produce visual artifacts such as floor penetration (as measured by the Floor Penetration metric in tables). This means that the network occasionally misses or ignores the PC context. This can

happen due to the noise presented in the pointcloud data and large distances between the points, especially in untextured regions like floors or walls. One way to improve the performance here is to use the more advanced point cloud/mesh reconstruction solution, potentially using the depth sensor (*e.g.* depth-based fusion [IKH+11]). Another way is to use a more advanced point cloud encoder; such an encoder can be trained on a different task, *e.g.*, point-to-mesh [CPM+20]. Note that we only capture static point clouds and do not yet handle dynamic environment changes such as opening doors, moving a chair, etc. – this is a great future work direction.

Our method is not aware of the shape of the body and, therefore, does not correct self-interpenetration of body parts, which can happen sometimes. That can be fixed during the postprocessing stage with self-contact optimization methods like TUCH [MOT+21]. Another problem that affects the visual quality is motion jitter, which can be observed mostly during online low-latency inference – this can be smoothed during motion post-processing. However, we decided not to apply the smoothing to show the raw performance of the method.

# Chapter 7

# Conclusions and Future Work

This dissertation and publications presented here study human motion capturing and generation from an aspect of egocentric wearable systems. This chapter provides a summary of contributions, hints at the connections, and highlights common ideas between the works presented here. It also discusses the potential future directions and societal impact of the research.

## 7.1 Conclusions

The works presented in this thesis demonstrate the potential of wearable systems for human motion recovery and interaction understanding. These methods are making an impact in the fields of telepresence, AR/VR, and robotics, and their usefulness will grow further in the future. As advancements in mobile computing continue, it will become easier to develop lighter, more powerful devices that are comfortable for users to wear and more adaptable for mobile platforms. This progress will ease the data collection, enabling the development of algorithms for motion generation and scene understanding from an egocentric perspective.

This thesis presents three publications in the fields of human motion generation and human motion and human-object interaction capturing from wearable sensors. In all works, we study different aspects of a proposed combination of camera and IMUs, like capturing stability, the ability to track human-object interactions, and the potential for miniaturization. These contributions are unified by a common goal – to expand the boundaries of what can be achieved with information constrained by an egocentric perspective.

Chapter 4 presents the first system capable of recovering full human body motion registered within large 3D scenes from wearable sensors only. The system combines advantages of global camera self-localization and inertial pose estimation, removing the

drift typical for IMU-based systems, and reducing the noise of camera localization. HPS enabled large-scale motion recording and allowed us to gather the HPS dataset – a collection of more than 3 hours of motion captured in large outdoors and indoors scenes. This is a unique result which is nearly impossible to achieve with conventional external camera motion capturing setups. The dataset has been made publicly available and has established itself as a benchmark in multiple egocentric localization and human motion generation studies [YZH+23, JSM+23, YLK+24].

HPS marks a pivotal moment in the field, laying the foundation for a new direction in egocentric motion capture and understanding. As the first work of its kind, HPS inspired the development of numerous subsequent systems and datasets, *e.g.* [DLW+22, YZH+23, ZMZ+22, MYH+24, YLK+24]. These follow-up efforts have built upon the capabilities introduced by HPS, driving advancements in egocentric motion capturing, generation, and understanding. Now, this emerging direction is rapidly gaining momentum and attracting increasing attention from the research community. The growing number of works and datasets in this area highlights its importance and potential, and we expect the field to continue expanding in the future.

In the following chapter, we extend the idea of wearable sensors capturing to human-object interaction. The iReplica system, presented in Chapter 5, is the first to capture both human and object they interact with using only the user-worn sensors, without relying on any additional external trackers. While iReplica relies on the insights gained from HPS, tracking objects presents several unique challenges, such as objects leaving the egocentric view and the need for more precise motion tracking. To address them, several innovations had to be made. First, we found that the contacts can be predicted solely from the body motion. Based on this insight, we developed and trained a method for the pose-based contact prediction, which reliably predicts human-object interaction timings. Second, we improved body localization upon HPS to match the precision required to capture dynamic interactions. Lastly, we designed an object motion inference pipeline based on predicted contacts. This made it possible to track object position changes without external input.

As the pioneering work in this area, iReplica represents a significant step forward in human-object interaction capturing. By leveraging egocentric data, it eliminates the need for external cameras or object-mounted sensors, bringing the scalability of HPS to the interaction tracking domain. To train and evaluate it, we collected two unique datasets of human-object interactions totaling over 3 hours of motion data, contact timestamps, egocentric videos, and more. To support further research in this area, we have made both the iReplica implementation and the collected datasets publicly available.

We further continue to explore the capabilities of wearable setups in Chapter 6 and present HMD$^2$ – a generative model for human motion conditioned on the data from a head-mounted device equipped with cameras and inertial sensors. This model stands in between the reconstruction and generation – it manages to follow the ground truth motion if it can be inferred from the information given and generate plausible motion otherwise. This is achieved through a novel approach: conditioning the generative motion diffusion network on a multi-modal input of camera, head motion, and SLAM scene points. As a second contribution, the new autoregressive inference scheme allows the model to generate long and uninterrupted sequences of motion with low latency between the input and output, making it suitable for real-time applications.

While the model has to deal with a very restricted input coming from a single point of view, HMD$^2$ demonstrates that a combination of several data modalities and a generative framework results in a realistic motion reconstruction even when using an extremely compact wearable capturing system. Such a user-friendly setup has direct applications in AR/VR motion tracking, providing a practical use case for the system today rather than as a prospect for the future.

Each system introduced in this thesis – HPS, iReplica, and HMD$^2$ – offers a distinct advancement, from large-scale human motion capture to precise human-object interaction tracking and generative motion modeling. Together, they establish a solid foundation for future research and open up exciting opportunities for applications.

## 7.2 Key Insights

At the beginning of this PhD, we set out to explore the potential of wearable systems for human motion capture and analysis with a grand goal to make it possible to recover human motion in the real world without the need for external cameras or markers. We did not know what to expect at first – which techniques and capturing methods would be successful and which would fail, but as the research progressed, a few fundamental findings emerged, shaping the direction of this work. The presented publications share these key insights, which we summarize below.

**A combination of the head camera and IMUs provides broad capturing capabilities.** The first key insight comes from the choice of the wearable capturing system itself, the sensors involved in it, and their layout. In Chapter 4, we present HPS - a human pose estimation and localization system from wearable sensors, namely the head-mounted camera looking outwards and inertial sensors mounted on the limbs and torso. With HPS, we prove that it is possible to localize the human in the scene using only body-mounted

sensors in the chosen configuration. Subsequent works, including Nymeria [MYH+24], coauthored by the author of this thesis, further reinforce our vision. Nymeria introduces a dataset of 300 hours of human motion captured from an egocentric device in large-scale environments, further validating the scalability and utility of wearable systems for motion capture. Building on this foundation, Chapter 5 extends the capabilities of wearable systems to human-object interactions. The iReplica method demonstrates that these systems are not only capable of localizing the user but can also track the objects they interact with, significantly broadening the scope of wearable capturing systems. Additionally, Chapter 6 demonstrates the potential of extremely portable versions of this setup, featuring the same types of sensors but reduced down to a single head-mounted device. This evolution towards minimalistic setups signifies a step forward in making wearable motion capture systems more accessible and comfortable for the average user.

**Sensor fusion is key to resolving ambiguities.** One important insight for dealing with such an underconstrained problem as egocentric motion reconstruction is a fusion of multiple sources of information. In Chapter 4, HPS demonstrates that joint optimization of IMU-inferred body motion, global camera-self localization, and scene pointcloud leads to a more stable and reliable solution compared to treating these information sources independently. In Chapter 5, iReplica shows that with deeper scene understanding, it is also possible to extend the capturing to the objects the user interacts with. The advantages of multi-source information fusion are yet again proven in Chapter 6 by $\text{HMD}^2$, which attends to local SLAM point clouds, head camera image, and head position simultaneously to generate body motion that is both realistic and contextually appropriate. These works collectively highlight the power of multi-source information fusion in wearable systems.

**Learning from human behavior helps to improve the quality.** Learned priors can substantially improve both capturing and generation. In Chapter 5, we demonstrate that a learned model can reliably predict the timing of human-object interactions for various motion classes based solely on body pose, which is crucial for capturing realistic object motion from wearables. This is achieved by training our method on a dataset of interactions collected during this project. Chapter 6 presents a generative model that is trained on hundreds of hours of human performance. With all sensors mounted on the head, situations frequently arise where input information is insufficient for motion reconstruction. In such cases, the strong generative priors of the model play a critical role, enabling it to produce realistic motion based on prior context and scene understanding.

**Datasets are critical for driving progress.** As discussed in the previous insight, the practical effectiveness of human motion capturing and generation methods often rely on

priors, which are, in turn, influenced by the quality, variability, and scale of data. In Chapter 4, we gather a dataset of more than 3 hours of human motion in large scenes. This dataset proved essential for advancing the motion generation methods and establishing benchmarks for future research [MPKPM24, YZH+23, ZMZ+22]. In Chapter 5, we collect a dataset of several hours of interactions with various objects and demonstrate that such a dataset can be an important part of building a human-object interaction capturing system. Additionally, Chapter 6 demonstrates that capturing motion at the scale of the Nymeria dataset enables the training of robust generative models capable of addressing heavily underconstrained tasks.

Developing these insights brings us closer to understanding the potential of wearable systems in human motion capture and analysis. Each work builds upon these foundational ideas, collectively pushing the boundaries of what is achievable with wearable technology in real-world scenarios.

## 7.3 Future work

The algorithms of motion capture from wearable setups have direct real-world applications. Examples include a personal assistant that understands your surroundings and past actions, offering context-aware suggestions and answering questions; a robot capable of forecasting plausible actions and motions based on the current scene layout; a VR meeting room where each person's motion is reconstructed from the VR helmet data, improving the realism and immersion. The potential applications are vast, and the works discussed in this thesis serve as a fundamental step toward implementing them. However, there are still challenges to overcome before these applications can be fully realized. This section presents the possible future directions that can be built on top of the presented contributions.

**Human-object interaction capturing for smaller objects.** HPS and iReplica showed the ability to capture the human-scene and human-object interactions with a precision high enough for realistic interactions with bigger pieces of furniture, such as sofas, doors, tables, and so on. But many applications, such as robotics, would greatly benefit from the ability to interact with the smaller everyday objects as well, like cups, keys, cutlery, *etc*. This challenges the existing systems because it would require much higher localization precision for both the subject in the scene and the object they operate with. Recent advancements in camera self-localization ([CEG+21, JCF23]) and object localization ([HSW+22]) show the potential for the direction. However, these algorithms have not yet been applied to human-object interaction tasks, so problems may arise with integrating

the results of both systems.

**Enhancing the evaluation of motion realism.** The works presented in this thesis demonstrate the ability to capture human motion and interactions with objects. However, the realism of the captured motion is not fully evaluated. The realism of motion is a complex concept that includes motion plausibility, naturalness, and the context-awareness of the actions. Hence, it cannot be measured using a simple distance error metric. Currently, realism can be assessed by designing a user study similar to the one we conducted in iReplica (Sec. 5.4.4), but this approach is not scalable. Some works attempt to solve this problem by comparing the generated motion distribution to the ground truth in a special latent space [GZW+20]. We used similar protocols in Chapter 6 to compare our results with the baselines. However, these metrics are heavily dependent on a latent space design and might not always correlate to the visual plausibility of motion, so evaluation of the realism of the motion is still an open question and is a promising direction for future research.

**Dynamic scenes capturing in the unknown environment.** iReplica method, discussed in Chapter 5 proves the possibility of human-object interaction capturing in the prescanned interactive scene with labeled objects. While this by itself has applications, the natural question is whether we can reduce the amount of prior information needed before the capturing. Given the growing capabilities of foundational image models, like DINOv2 [ODM+23] and instance segmentation networks [CS22, KMR+23], one could think of a capturing system capable of producing the same results without the need of labeling the information about the object, instead relying on learned methods to identify and correctly segment the object and determine its degrees of freedom. Given the increasing affordability of the wearable eye-tracking hardware, the problem of unknown object detection can be solved by using the advancements in gaze-based detection [WSZK20, WFZK22] where the information about the object location is inferred from the user's gaze direction.

**Extending sparse sensors capturing to human-object interactions.** $HMD^2$ shows the possibility of closely following human motion while operating with the data from the head-mounted sensors only. Yet, it does not reconstruct the objects the user interacts with, creating a potential for future improvement. As shown in iReplica and other works, *e.g.* Object Pop-Up [PMCPM23], it is possible to infer the object position from the body interaction alone. Additional information about the object can be acquired from the visual input: the class of the object and its partial reconstruction can be extracted if the person takes a look at the object before interacting. By fusing visual information and

inferred object motion together, it may be possible to recover the object model and pose in the same way HMD$^2$ reconstructs the body motion.

**Real-time on-device human motion recovery.** The works presented in this thesis demonstrate impressive capabilities in human motion capturing and generation. However, they all require, at least in part, processing on an external device, which the user cannot wear. In practice, it would be more beneficial if the algorithm could be executed on a portable device, closing the loop and making the wearable capturing system completely standalone. In HPS and iReplica, the major reason for external processing is the camera self-localization pipeline, which requires an external GPU for computation. It could be replaced by recent methods in visual-inertial odometry [RACC20, CEG+21, Pro23] that demonstrated portable solutions for localization pipelines with reliable precision. While iReplica and HPS could be optimized this way, HMD$^2$ would still require external processing despite having a less resource-intensive localization pipeline. The reason is the generative diffusion model, which requires a powerful GPU to maintain the real-time processing speed. Optimizing the inference of diffusion models is a hot topic in the scientific community nowadays, and several breakthroughs have been made in this direction, including reducing the number of reverse diffusion steps [KPS+24] and memory footprint [ZJD+24]. Together with the development of faster and more efficient mobile hardware, these advancements create the possibility of optimizing the current model for mobile device requirements.

**Human-to-robot skills transfer.** Wearable motion capture systems offer a promising approach to transferring human skills to robots by enabling large-scale, real-world motion data collection. Unlike traditional motion capture setups, they provide natural demonstrations from which robots can learn. Recent works on imitation learning from humans [LCK+23, CJC+24] suggest that using human motion datasets can significantly improve robot performance in various tasks. Notably, there is considerable potential in developing robot-object interaction skills from the data collected by wearable systems. By combining human-object interaction capture from iReplica with advances in robot object localization from human input [WFKZ23] and object manipulation learning [CWYL24], it is possible to create a system that can learn complex manipulation tasks from human demonstrations. This approach could be particularly useful in applications where the robot has to interact with many different objects in the environment, such as in household chores or industrial settings.

## 7.4 Privacy considerations

Since wearable sensors are body-mounted devices, they are typically worn for extended periods, often without consideration of turning them off. This raises important privacy concerns and necessitates measures to prevent potential data leakage. There are several strategies to address these concerns, each orthogonal to the others. One approach is to allow the device to only work in certain scenarios, like the user voice command, and equip the indicator that signals about the device activity. Another solution is to perform all the computations on the device so that the data never leaves it. While this is not always feasible, in such cases, we can reduce the data steam by precomputing the descriptive features, such as CLIP, and avoid sending the raw frames. Alternatively, we can use special descriptors designed with privacy in mind [SSK+19, DSSP21] to reduce the risk of leaking sensitive information about the user's surroundings.

Our methods use videos of human performance, both first-person and third-person view, and body shape and motion data, which are collected by us. We ensured that the subjects understood the ways the data would be used, and we obtained written consent from all the participants. Additionally, we anonymized the collected data by storing the motion and shape data in a depersonalized format and blurring the faces in the captured videos.

# Abbreviations

| | |
|---|---|
| DOF | Degrees Of Freedom |
| FID | Fréchet inception distance |
| GPU | Graphics Processing Unit |
| GT | Ground Truth |
| HMD | Head-Mounted Device |
| HOI | Human-Object Interaction |
| ICP | Iterative Closest Point |
| IE | Interactive Environment |
| IMU | Inertial Measurement Unit |
| MLP | Multi-Layer Perceptron |
| MPJPE | Mean Per-Joint Positional Error |
| MPS | Machine Perception Service |
| PC | Point Cloud |
| PE | Positional Error |
| RANSAC | RAndom SAmple Consensus |
| RGB(D) | Red, Green, Blue (and Depth) |
| SfM | Structure from Motion |
| SLAM | Simultaneous Localization and Mapping |
| VLAD | Vector of Locally Aggregated Descriptors |

# Bibliography


[AAA+23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[AGT+16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[ANBH23] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–20, 2023.

[APMTM19] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2293–2303. IEEE, Oct 2019.

[BCC+18] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2560–2568, 2018.

[BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[BR18] Eric Brachmann and Carsten Rother. Learning Less is More - 6D Camera Localization via 3D Surface Regression. In *CVPR*, 2018.

[BR20] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *arXiv:2002.12324*, 2020.

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[BSAJ17] Bharat Lal Bhatnagar, Suriya Singh, Chetan Arora, and C.V. Jawahar. Unsupervised learning of deep feature representation for clustering egocentric actions. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1447–1453, 2017.

[BSTPM20] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, August 2020.

[BTTPM19] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[BXP+22] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15935–15946, 2022.

[CEG+21] Carlos Campos, Richard Elvira, Juan J. Gómez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[CEJ+23] Angela Castillo, Maria Escobar, Guillaume Jeanneret, Albert Pumarola, Pablo Arbeláez, Ali Thabet, and Artsiom Sanakoyeu. Bodiffusion: Diffusing sparse observations for full-body human motion synthesis. *ICCV*, 2023.

[CGL+19] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr. Real-time rgb-d camera pose estimation in novel scenes using a relocalisation cascade. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[CGM+20] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*. 2020.

[CJC+24] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[CLSF10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *ECCV*, 2010.

[CPM+20] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020.

[CS22] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[CTM+21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[CWYL24] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.

[CZW+17] Congqi Cao, Yifan Zhang, Yi Wu, Hanqing Lu, and Jian Cheng. Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[DCS+17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[DFW+21] Siyan Dong, Qingnan Fan, He Wang, Ji Shi, Li Yi, Thomas Funkhouser, Baoquan Chen, and Leonidas J Guibas. Robust neural routing through

space partitions for camera relocalization in dynamic indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8544–8554, 2021.

[DKP+23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023.

[DLW+22] Yudi Dai, Yitai Lin, Chenglu Wen, Siqi Shen, Lan Xu, Jingyi Yu, Yuexin Ma, and Cheng Wang. Hsc4d: Human-centered 4d scene capture in large-scale indoor-outdoor space using wearable imus and lidar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6792–6802, 2022.

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[DNMC20] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6608–6617, 2020.

[DSSP21] Mihai Dusmanu, Johannes L Schonberger, Sudipta N Sinha, and Marc Pollefeys. Privacy-preserving image features via adversarial affine subspace embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14267–14277, 2021.

[DSZ+22] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.

[DT05] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[ESG+23] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. Project Aria: A new tool for egocentric multi-modal AI research, 2023.

[FB81] M. Fischler and R. Bolles. Random Sampling Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of the ACM (CACM)*, 24:381–395, 1981.

[FFR11] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *2011 international conference on computer vision*, pages 407–414. IEEE, 2011.

[FTT+23] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[Gal90] Charles R Gallistel. *The organization of learning.* The MIT Press, 1990.

[GCM+24] 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.

[GJH+25] Vladimir Guzov, Yifeng Jiang, Fangzhou Hong, Gerard Pons-Moll, Richard Newcombe, C. Karen Liu, Yuting Ye, and Lingni Ma. Hmd$^2$: Environment-aware motion generation from single egocentric head-mounted device. In *International Conference on 3D Vision (3DV)*, 2025.

[GMSPM21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329. IEEE, jun 2021.

[GNK18] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018.

[GPPM24] Vladimir Guzov, Ilya A Petrov, and Gerard Pons-Moll. Blendify - python rendering framework for blender. *arXiv preprint arXiv:2410.17858*, 2024.

[GZW+20] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020.

[HAB20] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020.

[HBB+13] Thomas Helten, Andreas Baak, Gaurav Bharaj, Meinard Muller, Hans-Peter Seidel, and Christian Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *International Conf. on 3D Vision*, pages 279–286, 2013.

[HCTB19] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *Proceedings International Conference on Computer Vision*, pages 2282–2292. IEEE, October 2019.

[HCV+21] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021.

[HGK+24] Fangzhou Hong, Vladimir Guzov, Hyo Jin Kim, Yuting Ye, Richard Newcombe, Ziwei Liu, and Lingni Ma. Egolm: Multi-modal language model of egocentric motions. *arXiv preprint arXiv:2409.18127*, Sep 2024.

[HGT+21] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[HKPP20] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020.

[HKS17] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[HLON94] R.M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision (IJCV)*, 13(3):331–356, 1994.

[HPD+24] Bo Han, Hao Peng, Minjing Dong, Chang Xu, Yi Ren, Yixuan Shen, and Yuheng Li. Amd autoregressive motion diffusion. *AAAI*, 2024.

[HS+88] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[HSG+22] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[HSW+22] Xingyi He, Jiaming Sun, Yuang Wang, Di Huang, Hujun Bao, and Xiaowei Zhou. Onepose++: Keypoint-free one-shot object pose estimation without cad models. *Advances in Neural Information Processing Systems*, 35:35103–35115, 2022.

[HTBT22] Yinghao Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. Intercap: Joint markerless 3d tracking of humans and objects in interaction. In *Pattern Recognition: 44th DAGM German Conference, DAGM GCPR 2022, Konstanz, Germany, September 27–30, 2022, Proceedings*, pages 281–299. Springer, 2022.

[HTTM12] Michael B Holte, Cuong Tran, Mohan M Trivedi, and Thomas B Moeslund. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of selected topics in signal processing*, 6(5):538–552, 2012.

[HWL+23] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023.

[HYH+22] Chun-Hao P Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J Black. Capturing and inferring dense full-body human-scene contact. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13285, 2022.

[IBLM19] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7718–7727, 2019.

[IKH+11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction

and interaction using a moving depth camera. In *ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[IZFB09] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From Structure-from-Motion Point Clouds to Fast Location Recognition. In *CVPR*, 2009.

[JAG+21] Ara Jafarzadeh, Manuel López Antequera, Pau Gargallo, Yubin Kuang, Carl Toft, Fredrik Kahl, and Torsten Sattler. Crowddriven: A new challenging dataset for outdoor visual localization. In *ICCV*, 2021.

[JCF23] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17408–17419, June 2023.

[JDSP10] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[JG17] Hao Jiang and Kristen Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3509. IEEE, 2017.

[JS11] Eagle S. Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

[JSM+23] Jiaxi Jiang, Paul Streli, Manuel Meier, Andreas Fender, and Christian Holz. Egoposer: Robust real-time ego-body pose estimation in large scenes. *arXiv preprint arXiv:2308.06493*, 2023.

[JSQ+22] Jiaxi 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 *European Conference on Computer Vision*, pages 443–460. Springer, 2022.

[JSS18] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 8320–8329, 2018.

[KAB20] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KBJM18] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[KBP10] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-Form Solutions to Minimal Absolute Pose Problems with Known Vertical Direction. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[KMR+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[KPA+24] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 1334–1345. IEEE, 2024.

[KPS+24] Jonas Kohler, Albert Pumarola, Edgar Schönfeld, Artsiom Sanakoyeu, Roshan Sumbaly, Peter Vajda, and Ali Thabet. Imagine flash: Accelerating emu diffusion models with backward distillation. *arXiv preprint arXiv:2405.05224*, 2024.

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A Novel Parametrization of the Perspective-Three-Point Problem for a Direct

Computation of Absolute Camera Position and Orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[KTS+21] Taein Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. *arXiv preprint arXiv:2104.11181*, 2021.

[KWHG20] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[LCK+23] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.

[LGK20] Zhengyi Luo, S Alireza Golestaneh, and Kris M Kitani. 3d human motion estimation via motion compression and refinement. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[LGO+23] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Winter Conference on Applications of Computer Vision (WACV)*, January 2023.

[LHG+23] Diogo C. Luvizon, Marc Habermann, Vladislav Golyanik, Adam Kortylewski, and Christian Theobalt. Scene-aware 3d multi-human motion capture from a single camera. *Comput. Graph. Forum*, 42(2):371–383, 2023.

[LIYK22] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. *arXiv preprint arXiv:2206.09106*, 2022.

[LJ23] Jiye Lee and Hanbyul Joo. Locomotion-action-manipulation: Synthesizing human-scene interactions in complex 3d environments. *ICCV*, 2023.

[LJ24] Jiye Lee and Hanbyul Joo. Mocap everyone everywhere: Lightweight motion capture with smartwatches and a head-mounted camera. *Computer Vision and Pattern Recognition (CVPR)*, 2024.

[LJX+21] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021.

[LLW23] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17142–17151, 2023.

[LMC12] Karel Lebeda, Juan E. Sala Matas, and Ondřej Chum. Fixing the Locally Optimized RANSAC. In *British Machine Vision Conference (BMVC)*, 2012.

[LMR+15] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 2015.

[LS18] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. pages 2041–2050, Salt Lake City, UT, USA, 2018. IEEE.

[LSB+15] Simon Lynen, Torsten Sattler, Michael Bosse, Joel A Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, volume 1, page 1, 2015.

[LSHF12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. World-wide pose estimation using 3d point clouds. In *ECCV*, pages 15–29. Springer, 2012.

[LZA+20] Simon Lynen, Bernhard Zeisl, Dror Aiger, Michael Bosse, Joel Hesch, Marc Pollefeys, Roland Siegwart, and Torsten Sattler. Large-scale, real-time visual–inertial localization revisited. *The International Journal of Robotics Research*, 39(9):1061–1084, 2020.

[LZCVDP20] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020.

[MAMT15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *TRO*, 31(5):1147–1163, 2015.

[MAO+19] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9339–9347, 2019.

[MAPM20] Aymen Mir, Thiemo Alldieck, and Gerard Pons-Moll. Learning to transfer texture from clothing images to 3d humans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020.

[MFK16] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, 2016.

[MGT+19] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[Mic24] *Microsoft Azure Kinect*, accessed November 1, 2024. https://en.wikipedia.org/wiki/Azure_Kinect.

[Mix24] *Mixamo rigging and animation service*, accessed December 2, 2024. https://www.mixamo.com/.

[MLT88] Thalmann Magnenat, Richard Laperrière, and Daniel Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings of Graphics Interface'88*, pages 26–33. Canadian Inf. Process. Soc, 1988.

[MOT+21] Lea Muller, Ahmed AA Osman, Siyu Tang, Chun-Hao P Huang, and Michael J Black. On self-contact and human pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9990–9999, 2021.

[MPKPM24] Aymen Mir, Xavier Puig, Angjoo Kanazawa, and Gerard Pons-Moll. Generating continual human motion in diverse 3d scenes. In *2024 International Conference on 3D Vision (3DV)*, pages 903–913. IEEE, 2024.

[MVG+17] Charles Malleson, Marco Volino, Andrew Gilbert, Matthew Trumble, John Collomosse, and Adrian Hilton. Real-time full-body motion capture from video and imus. In *2017 Fifth International Conference on 3D Vision (3DV)*, 2017.

[MYA+24] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36, 2024.

[MYH+24] Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David S. Fosas, C. Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, and Richard Newcombe. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *European Conference on Computer Vision (ECCV)*, 2024.

[nav20] *Navvis M16*, accessed November 15, 2020. https://www.navvis.com/m6.

[ND21] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[NOT24] Ana Filipa Rodrigues Nogueira, Hélder P. Oliveira, and Luís F. Teixeira. Markerless multi-view 3d human pose estimation: a survey, 2024.

[ODM+23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[OLPM+18] Mohamed Omran, Christop Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model

based human pose and shape estimation. In *International Conf. on 3D Vision*, 2018.

[OWL19] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):1898–1912, 2019.

[PCG+19] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[PLPM20] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[PMBG+11] Gerard Pons-Moll, Andreas Baak, Juergen Gall, Laura Leal-Taixé, Meinard Muller, Hans-Peter Seidel, and Bodo Rosenhahn. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *Proceedings of the 2011 International Conference on Computer Vision (ICCV)*, pages 1243–1250, 2011.

[PMBH+10] Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 663–670. IEEE, 2010.

[PMCPM23] Ilya A Petrov, Riccardo Marin, Julian Chibane, and Gerard Pons-Moll. Object pop-up: Can we infer 3d objects and their poses from human interactions alone? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4726–4736, 2023.

[PMR11] Gerard Pons-Moll and Bodo Rosenhahn. *Model-Based Pose Estimation*, chapter 9, pages 139–170. Springer, 2011.

[PMY+23] Shaohua Pan, Qi Ma, Xinyu Yi, Weifeng Hu, Xiong Wang, Xingkang Zhou, Jijunnan Li, and Feng Xu. Fusing monocular images and sparse IMU signals for real-time human motion capture. In June Kim, Ming C. Lin, and Bernd Bickel, editors, *SIGGRAPH Asia 2023 Conference Papers, SA 2023, Sydney, NSW, Australia, December 12-15, 2023*, pages 116:1–116:11. ACM, 2023.

[Pro23] Project aria machine perception services, 2023. https://facebookresearch.github.io/projectaria_tools/docs/ARK/mps.

[Pro24] *Project Aria*, accessed November 12, 2024. https://www.projectaria.com.

[PSRB18] Monique Paulich, Martin Schepers, Nina Rudigkeit, and G. Bellusci. *Xsens MTw Awinda: Miniature Wireless Inertial-Magnetic Motion Tracker for Highly Accurate 3D Kinematic Applications*, 05 2018.

[PX23] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[PZZD18] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[RACC20] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

[RBH+21] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. pages 11468–11479, Montreal, QC, Canada, 2021. IEEE.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

[RHH+20] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6040–6049, 2020.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[RLL+23] Sigal Raab, Inbal Leibovitch, Peizhuo Li, Kfir Aberman, Olga Sorkine-Hornung, and Daniel Cohen-Or. Modi: Unconditional motion synthesis from diverse data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13873–13883, 2023.

[RLS07] Daniel Roetenberg, Henk Luinge, and Per Slycke. Moven: Full 6dof human motion tracking using miniature inertial sensors. *Xsen Technologies, December*, 2007.

[RLS09] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technol. BV Tech. Rep.*, 3, 01 2009.

[RRC+16] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):162, 2016.

[RSR15] Grégory Rogez, James S Supancic, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4325–4333, 2015.

[RTB17] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

[SBB10] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1):4–27, 2010.

[SC05] Aravind Sundaresan and Rama Chellappa. Markerless motion capture using multiple cameras. In *Computer Vision for Interactive and Intelligent Environment (CVIIE'05)*, pages 15–26. IEEE, 2005.

[SCSD19] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[SDMR20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks. In *CVPR*, 2020.

[SF16] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SGSF20] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020.

[SGXT20] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (ToG)*, 39(6):1–16, 2020.

[SGZ+13] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[Sho85] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[SLAL20] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. Metrabs: Metric-scale truncation-robust heatmaps for absolute 3d human pose estimation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2020.

[SLK17] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient effective prioritized matching for large-scale image-based localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1744–1756, 2017.

[SPBK20] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8623–8632, 2020.

[SPGS18] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic Visual Localization. In *CVPR*, 2018.

[SPS+11] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K Hodgins. Motion capture from body-mounted cameras. In *ACM Transactions on Graphics (TOG)*, volume 30, page 31. ACM, 2011.

[SSK+19] Pablo Speciale, Johannes L Schonberger, Sing Bing Kang, Sudipta N Sinha, and Marc Pollefeys. Privacy preserving image-based localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5493–5503, 2019.

[STKB23a] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023.

[STKB23b] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *ICLR*, 2023.

[SWJD23] Yi Shi, Jingbo Wang, Xuekun Jiang, and Bo Dai. Controllable motion diffusion model. *arXiv preprint arXiv:2306.00416*, 2023.

[SZKS19] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.

[SZPF16] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View Selection for Unstructured Multi-View Stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[TAL+22] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *European Conference on Computer Vision*, pages 572–589. Springer, 2022.

[TAP+20] Denis Tome, Thiemo Alldeick, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando de la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Oct 2020.

[TBP19] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4511–4520, 2019.

[TCBT22] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13263–13273, 2022.

[TCL23] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, June 2023.

[TGBT20] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 581–600. Springer, 2020.

[TGM+17] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video

and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017.

[TMT+20] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Danial Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, Fredrik Kahl, and Torsten Sattler. Long-Term Visual Localization Revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1, 2020.

[TOS+21] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomás Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(4):1293–1307, 2021.

[TSOP15] Akihiko Torii, Josef Sivic, Masatoshi Okutomi, and Tomás Pajdla. Visual place recognition with repetitive structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(11):2346–2359, 2015.

[TWZ20] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 197–212. Springer, 2020.

[VAV+07] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical motion capture in everyday surroundings. *ACM Transactions on Graphics (TOG)*, 26(3):35, 2007.

[vMHB+18] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conf. on Computer Vision*, sep 2018.

[vMPMR16] T von Marcard, G. Pons-Moll, and B. Rosenhahn. Human pose estimation from video and IMUs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(8):1533–1547, 2016.

[vRBPM17] Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), Proceedings of the 38th*

*Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 349–360, 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WCL+22] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. *Advances in Neural Information Processing Systems*, 35:14959–14971, 2022.

[WFKZ23] Daniel Weber, Wolfgang Fuhl, Enkelejda Kasneci, and Andreas Zell. Multiperspective teaching of unknown objects via shared-gaze-based multimodal human-robot interaction. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 544–553, 2023.

[WFZK22] Daniel Weber, Wolfgang Fuhl, Andreas Zell, and Enkelejda Kasneci. Gaze-based object detection in the wild. In *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, pages 62–66. IEEE, 2022.

[WLF+24] Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott L. Delp, and C. Karen Liu. Diffusionposer: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 2513–2523. IEEE, 2024.

[WLNM21] Yu-Shiang Wong, Changjian Li, Matthias Niessner, and Niloy J. Mitra. Rigidfusion: Rgb-d scene reconstruction with rigidly-moving objects. *Computer Graphics Forum*, 40(2), 2021.

[WLX+23] Jian Wang, Diogo Luvizon, Weipeng Xu, Lingjie Liu, Kripasindhu Sarkar, and Christian Theobalt. Scene-aware egocentric 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13031–13040, 2023.

[WPY+19] He Wang, Sören Pirk, Ersin Yumer, Vladimir G Kim, Ozan Sener, Srinath Sridhar, and Leonidas J Guibas. Learning a generative model for multi-step human-object interactions from videos. In *Computer Graphics Forum*, volume 38, pages 367–378. Wiley Online Library, 2019.

[WRL+22] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20460–20469, 2022.

[WSG+20] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3d camera relocalization in changing indoor scenes. In *European Conference on Computer Vision*, pages 467–487. Springer, 2020.

[WSZK20] Daniel Weber, Thiago Santini, Andreas Zell, and Enkelejda Kasneci. Distilling location proposals of unknown objects through gaze information for human-robot interaction. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11086–11093. IEEE, 2020.

[WY21] Zhenzhen Weng and Serena Yeung. Holistic 3d human and scene mesh estimation from single view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 334–343, 2021.

[WZ18] Ya Wang and Andreas Zell. Improving feature-based visual slam by semantics. In *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pages 7–12. IEEE, 2018.

[XCZ+19] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. $Mo^2Cap^2$ : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.

[XJMS21] Xiang Xu, Hanbyul Joo, Greg Mori, and Manolis Savva. D3d-hoi: Dynamic 3d human-object interactions from videos. *arXiv preprint arXiv:2108.08420*, 2021.

[Xse24] accessed November 25, 2024. https://www.xsens.com/products/mvn-analyze.

[XZH+18] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[YK18] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 735–750, 2018.

[YK19] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[YKL21] Dongseok Yang, Doyeon Kim, and Sung-Hee Lee. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. In *Computer Graphics Forum*, volume 40, pages 265–275. Wiley Online Library, 2021.

[YLK+24] 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 Transactions on Graphics (TOG)*, 43(6):1–12, 2024.

[YLZ21] Chenhao Yang, Yuyi Liu, and Andreas Zell. Learning-based camera re-localization with domain adaptation via image-to-image translation. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1047–1054. IEEE, 2021.

[YMO+15] H. Yonemoto, K. Murasaki, T. Osawa, K. Sudo, J. Shimamura, and Y. Taniguchi. Egocentric articulated pose tracking for action recognition. In *International Conference on Machine Vision Applications (MVA)*, 2015.

[YTY+23] Wenjie Yin, Ruibo Tu, Hang Yin, Danica Kragic, Hedvig Kjellström, and Mårten Björkman. Controllable motion synthesis and reconstruction with autoregressive diffusion models. *arXiv preprint arXiv:2304.04681*, 2023.

[YZH+22] 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. pages 13157–13168, New Orleans, LA, USA, 2022. IEEE.

[YZH+23] 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 Transactions on Graphics (TOG)*, 42(4):1–17, 2023.

[YZX21] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

[ZBLD19] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11776–11785, 2019.

[ZBS+22] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*. Springer, October 2022.

[ZBS+25] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Ilya Petrov, Vladimir Guzov, Helisa Dhamo, Eduardo Pérez Pellitero, and Gerard Pons-Moll. Force: Dataset and method for intuitive physics guided human-object interaction. In *International Conference on 3D Vision (3DV)*, 2025.

[ZDC+23] Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. Emdm: Efficient motion diffusion model for fast, high-quality motion generation. *arXiv preprint arXiv:2312.02256*, 2023.

[ZJD+24] Yang Zhang, Er Jin, Yanfei Dong, Ashkan Khakzar, Philip Torr, Johannes Stegmaier, and Kenji Kawaguchi. Effortless efficiency: Low-cost pruning of diffusion models. *arXiv preprint arXiv:2412.02852*, 2024.

[ZLAH23] Zihan Zhang, Richard Liu, Kfir Aberman, and Rana Hanocka. Tedi: Temporally-entangled diffusion for long-term motion synthesis. *arXiv preprint arXiv:2307.15042*, 2023.

[ZMZ+22] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. Egobody: Human body shape and motion of interacting people from head-mounted devices. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 180–200. Springer, Oct 2022.

[ZPJ+20] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *European Conference on Computer Vision (ECCV)*, 2020.

[ZPVBG01] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378, 2001.

[ZSG+24] Xiaohan Zhang, Sebastian Starke, Vladimir Guzov, Zhensong Zhang, Eduardo Pérez Pellitero, and Gerard Pons-Moll. Scenic: Scene-aware semantic navigation with instruction-guided control. *arXiv preprint arXiv:2412.15664*, Dec 2024.

[ZSH+23] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. *arXiv preprint arXiv:2303.17076*, 2023.

[ZWZ+22] Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, , and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *European conference on computer vision (ECCV)*, October 2022.

[ZWZ+24] 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. pages 2209–2219, Seattle, WA, USA, 2024. IEEE.

[ZYL+18] Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Quionghai Dai, Lu Fang, and Yebin Liu. Hybridfusion: Real-time performance capture using a sin-

gle depth sensor and sparse imus. In *European Conference on Computer Vision (ECCV)*, 2018.

[ZYM+22] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, C Karen Liu, and Leonidas J Guibas. Gimo: Gaze-informed human motion prediction in context. In *European Conference on Computer Vision*, pages 676–694. Springer, 2022.

[ZZB+21] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11343–11353, 2021.

[ZZW+23] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. DIMOS: Synthesizing diverse human motions in 3d indoor scenes. In *International Conference on Computer Vision (ICCV)*, 2023.