Whole-Body Motion Capture and Beyond: From Model-Based Inference to Learning-Based Regression

Whole-Body Motion Capture and Beyond: From Model-Based Inference to Learning-Based Regression

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. Yinghao Huang
aus Luoyang, China

Tübingen
2022

Tag der mündlichen Qualifikation:	06.12.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Michael J. Black
2. Berichterstatter:	Prof. Dr. Helge Rhodin

To my parents

Abstract

Though effective and successful, traditional marker-less Motion Capture (MoCap) methods suffer from several limitations: 1) they presume a character-specific body model, thus they do not permit a fully automatic pipeline and generalization over diverse body shapes; 2) no objects humans interact with are tracked, while in reality interaction between humans and objects is ubiquitous; 3) they heavily rely on a sophisticated optimization process, which needs a good initialization and strong priors. This process can be slow. We address all the aforementioned issues in this thesis, as described below.

Firstly we propose a fully automatic method to accurately reconstruct a 3D human body from multi-view RGB videos, the typical setup for MoCap systems. We pre-process all RGB videos to obtain 2D keypoints and silhouettes. Then we fit the SMPL body model into the 2D measurements in two successive stages. In the first stage, the shape and pose parameters of SMPL are estimated frame-wise sequentially. In the second stage, a batch of frames are refined jointly with an extra DCT prior. Our method can naturally handle different body shapes and challenging poses without human intervention.

Then we extend this system to support tracking of rigid objects the subjects interact with. Our setup consists of 6 Azure Kinect cameras. Firstly we pre-process all the videos by segmenting humans and objects and detecting 2D body joints. We adopt the SMPL-X model here to capture body and hand pose. The model is fitted to 2D keypoints and point clouds. Then the body poses and object poses are jointly updated with contact and interpenetration constraints. With this approach, we capture a novel human-object interaction dataset with natural RGB images and plausible body and object motion information.

Lastly, we present the first practical and lightweight MoCap system that needs only 6 IMUs. Our approach is based on Bi-directional RNNs. The network can make use of temporal information by jointly reasoning about past and future IMU measurements. To handle the data scarcity issue, we create synthetic data from archival MoCap data. Overall, our system runs ten times faster than traditional optimization-based methods, and is numerically more accurate. We also show it is feasible to estimate which activity the subject is doing by only observing the IMU measurement from a smartwatch worn by the subject. This not only can be useful for a high-level semantic understanding of the human behavior, but also alarms the public of potential privacy concerns.

In summary, we advance marker-less MoCap by contributing the first automatic yet accurate system, extending the MoCap methods to support rigid object tracking, and proposing a practical and lightweight algorithm via 6 IMUs. We believe our work makes marker-less and IMUs-based MoCap cheaper and more practical, thus closer to end-users for daily usage.

Kurzfassung

Herkömmliche markerlose Motion Capture (MoCap)-Methoden sind zwar effektiv und erfolgreich, haben aber mehrere Einschränkungen: 1) Sie setzen ein charakterspezifisches Körpermodell voraus und erlauben daher keine vollautomatische Pipeline und keine Verallgemeinerung über verschiedene Körperformen; 2) es werden keine Objekte verfolgt, mit denen Menschen interagieren, während in der Realität die Interaktion zwischen Menschen und Objekten allgegenwärtig ist; 3) sie sind in hohem Maße von ausgeklügelten Optimierungen abhängig, die eine gute Initialisierung und starke Prioritäten erfordern. Dieser Prozess kann sehr zeitaufwändig sein.

In dieser Arbeit befassen wir uns mit allen oben genannten Problemen. Zunächst schlagen wir eine vollautomatische Methode zur genauen 3D-Rekonstruktion des menschlichen Körpers aus RGB-Videos mit mehreren Ansichten vor. Wir verarbeiten alle RGB-Videos vor, um 2D-Keypoints und Silhouetten zu erhalten. Dann passen wir modell in zwei aufeinander folgenden Schritten an die 2D-Messungen an. In der ersten Phase werden die Formparameter und die Posenparameter der SMPL nacheinander und bildweise geschätzt. In der zweiten Phase wird eine Reihe von Einzelbildern gemeinsam mit der zusätzlichen DCT-Priorisierung (Discrete Cosine Transformation) verfeinert. Unsere Methode kann verschiedene Körperformen und schwierige Posen ohne menschliches Zutun verarbeiten.

Dann erweitern wir das MoCap-System, um die Verfolgung von starren Objekten zu unterstützen, mit denen die Testpersonen interagieren. Unser System besteht aus 6 RGB-D Azure-Kameras. Zunächst werden alle RGB-D-Videos vorverarbeitet, indem Menschen und Objekte segmentiert und 2D-Körpergelenke erkannt werden. Das SMPL-X-Modell wird hier eingesetzt, um die Handhaltung besser zu erfassen. Das SMPL-X-Modell wird in 2D-Keypoints und akkumulierte Punktwolken eingepasst. Wir zeigen, dass die Körperhaltung wichtige Informationen für eine bessere Objektverfolgung liefert. Anschließend werden die Körper- und Objektposen gemeinsam mit Kontakt- und Durchdringungsbeschränkungen optimiert. Mit diesem Ansatz haben wir den ersten Mensch-Objekt-Interaktionsdatensatz mit natürlichen RGB-Bildern und angemessenen Körper- und Objektbewegungsinformationen erfasst.

Schließlich präsentieren wir das erste praktische, leichtgewichtige MoCap-System, das nur 6 Inertialmesseinheiten (IMUs) benötigt. Unser Ansatz basiert auf bi-direktionalen rekurrenten neuronalen Netzen (Bi-RNN). Das Netzwerk soll die zeitliche Abhängigkeit besser ausnutzen, indem es vergangene und zukünftige Teilmessungen der IMUs zusammenfasst. Um das Problem der Datenknappheit zu lösen, erstellen wir synthetische Daten aus archivierten MoCap-Daten. Insgesamt läuft unser System 10 Mal schneller als

die Optimierungsmethode und ist numerisch genauer. Wir zeigen auch, dass es möglich ist, die Aktivität der Testperson abzuschätzen, indem nur die IMU-Messung der Smartwatch, die die Testperson trägt, betrachtet wird.

Zusammenfassend lässt sich sagen, dass wir die markerlose MoCap-Methode weiterentwickelt haben, indem wir das erste automatische und dennoch genaue System beisteuerten, die MoCap-Methoden zur Unterstützung der Verfolgung starrer Objekte erweiterten und einen praktischen und leichtgewichtigen Algorithmus mit 6 IMUs vorschlugen. Wir glauben, dass unsere Arbeit die markerlose MoCap billiger und praktikabler macht und somit den Endnutzern für den täglichen Gebrauch näher bringt.

Acknowledgments

First and foremost, I would like to express my sincere and deep gratitude to my Ph.D. advisor, Director Michael J. Black. It is Michael who recruited me six years ago, thus opening the door to the fantastic 3D Human Modeling world for me. Over the whole doctoral period, he not only taught me everything about how to do great, solid, and influential research in the area of the 3D human body understanding, but also told me extra aspects I need to learn or improve to become a better person in daily life. I have always been highly impressed by the fact that he is super insightful towards sensing and conceiving promising research topics while also quite familiar with almost all the technical details. He is a genuine researcher, a skillful and mature educator, and a successful entrepreneur in my eyes. I was fortunate to work with him during my Ph.D. stage.

It is a great honor for me to have Professor Hendrik P. A. Lensch, Professor Gerard Pons-Moll, and Professor Helge Rhodin as my community members. Working in the areas of Computer Vision, Computer Graphics and 3D Body Modeling, I cannot know for sure how many times I check research papers (co-)authored by them every day, and reading their publications helps me a lot in finding inspiration for new ideas and technical details for specific problems. Their willingness in spending precious time reviewing my thesis and attending my defense means a lot to me, and having personal contact with them during the graduation is like a dream come true. It is quite exhaustive to go through Ph.D. graduation, and things become enjoyable and smooth with their help and support.

I also want to thank other researchers I got the chance to work with (in order of time of first collaboration): Federica, Angjoo, Christoph, Peter, Javier, Ijaz, Gerard, Otmar, Dimitris, Omid, Manuel, Emre, Tony, Nikolaos, Yuanlu and Zeng. It would be way much harder for me to finish this thesis without their help. I mainly gained knowledge of how SMPL works under the hood from Federica, Angjoo, Javier, and Christoph. Gerard taught me how to systematically verify the hypotheses and how to pinpoint where the problem is. The discussion with him about how Dyna works strengthened my understanding of registration a lot. Manuel and Emre are great friends and collaborators to me. I will never forget the days and nights we stayed together all the time to figure out ways to get network training and real-time demo ready. I still remember the warm and comforting words from Peter when I felt puzzled about research. The words encouraged me a lot. Christoph is the most excellent programmer among all the senior researchers I know. He is always so principled, doing things at his own pace. Dimitris is always glad to lend his hand. No matter when I have any problems in research or life, he is always there to help. I would also like to thank Tony for hosting me during my internship at Facebook Reality Labs Sausalito. Besides being a great researcher in 3D humans, Tony

also has a lot of industry experience. Talking with him about his colorful past was very pleasing. My life at Sausalito would not have been so enjoyable without the support and guidance from Nikolaos, Zeng, Mir Rayat, Mathias, Rania, Yuanlu, Chengcheng, and Yijing. We shared our experiences and learned from each other.

The discussion with other group members inside PS and visiting interns has always been fruitful. Jonas and Anurag are the experts on dense and structured pixel-wise prediction. Timo and Jinlong know everything about mesh processing, and talking with them is a lot of fun and fruitful. Partha is so great at Machine Learning theory that he always gives the correct answer whenever one has a mathematical question. Ahmed is the new 3D human model building specialist, and I am always profoundly touched by his pure friendship. Siyu was so energetic and active. Her optimistic and risk-taking attitude touched me a lot. And her graph-theoretic way of solving pedestrian tracking is beautiful. Vassilias, Nikos, and Yao are familiar with the complex CUDA programming. Qianli, Mohamed, and Muhamed are always good at finding elegant and solid solutions to practical problems. Yan holds expertise in both traditional Machine Learning methods and new Deep Learning ones. Shunsuke is so productive in idea conception and implementation. His deep understanding of 3D Human Digitization impresses me a lot. I also would like to thank Omid and Victoria for sharing with me their knowledge about hand and face modeling. I am sure that my Ph.D. life will be much harder without these people to learn from and discuss with. Besides research, hanging out with Timo, Soubhik, Partha, Ahmed, Marilin, Lea, and many others is very pleasing. I am proud that I have been part of the PS research group.

The support and help I got from Melanie, Nicole, Johanna, and Benjamin was priceless. They can always get the administration and IT issues resolved in no time. It is a pity that we cannot see Rocko anymore. As the loving pet of the whole department, he used to wander around everywhere inside the office, bringing joy to us all. The snacks prepared and brought by Lee are so delicious, and I also enjoyed the party held at Michael's and Lee's apartment a lot. Kai, Yingjing, and Chuanfu from the developmental biology department gave me warm comfort when I faced the most challenging time. The gathering with them is always so relaxing, and playing Ping-Pong together is so much fun. Though located in different countries, Xiaolei, Mengdi, Zhiliang, and I continued our friendship beginning from the bachelor stage. Especially Xiaolei, who is so keen on mathematical foundations, enlightened my mind many times. I am sure we will continue this pure friendship in the future.

Last but not least, I would like to express my deepest thankfulness to my parents and sister. Under every circumstance, they support me wholeheartedly without any conservation. Sometimes they might not be so sure about whether I am making the right decision, but they still choose to believe in me and allow me to explore the future freely. Not until the end of my Ph.D. study did I realize how important my family is to me and what amount of love and support I have received and am still receiving from them. My parents are my heroes, and I hope I can be a person as great as they are.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Motion Capture, commonly abbreviated as MoCap, is the process of recording the movements of objects, animals, or people. Through this procedure, natural motions taking place inside the physical world can be faithfully converted into a digital form that is better suited for further editing and usage. As one of the fundamental tools for content creation, MoCap finds wide applications in film making, entertainment, healthcare, gaming, etc. Probably the best known MoCap system is Vicon[1], which has been adopted in the production of many award-winning movies. Other common MoCap technology providers include XSens[2], OptiTrack[3], and NeuroMoCap[4]. One snapshot of the Vicon system with a subject performing motions is shown in Figure 1.1.

Though highly effective and mature, all the existing solutions to MoCap suffer from some common limitations: the specialist devices are either time-consuming to set up or are intrusive to the actors, and sometimes even heavy manual labor is needed for data cleaning or other purposes. Taking the Vicon system for example, to use this system, firstly, workers need to arrange the special optical cameras in a large capture space. Then several optical markers are placed on the object to be captured. A problem that happens quite often in the data capture process is that markers are missing or wrongly categorized due to occlusion. To correct this issue, trained workers must manually correct these mismatches. The whole procedure is both time-consuming and tedious.

These considerations motivate Computer Vision and Computer Graphics researchers to search for more efficient and practical approaches to MoCap. An ideal MoCap system should satisfy three requirements: 1) It should be as accurate as possible. The captured digital motion should reflect the actual motion to a satisfactory extent; 2) It should yield results quickly enough. The users should not have to wait long to check and evaluate the performance. In cases where response speed is critical for user experience, like in VR/AR applications, even an interactive rate should be guaranteed; 3) It should

[1] https://www.vicon.com/
[2] https://www.xsens.com/
[3] https://optitrack.com/
[4] https://neuronmocap.com/

Figure 1.1: Illustration of the Vicon system inside the Perceiving Systems department of the Max Planck Institute for Intelligent Systems.

place minimal intrusiveness on the users. The ultimate goal is that the end-user activities should not be affected by the usage of the system at all. Marker-less MoCap (Xu et al., 2018a; Habermann et al., 2019, 2020, 2021a; Rhodin et al., 2016b), is one promising line of research aiming at all these targets mentioned above, where the subject needs to wear no extra accessories, thus has the full freedom to perform the desired activities naturally. Typically multiple cameras are required, and the cameras need to be calibrated to high accuracy beforehand.

Another direction also closely related to this thesis is MoCap from flexible and minimal-sized devices, like , the SIP method (Von Marcard et al., 2017), the AirCap method (Price et al., 2018; Saini et al., 2019), the EgoCap method (Rhodin et al., 2016a), and the Fly-con method proposed in (Nägeli et al., 2018). These approaches target getting accurate enough results at a relatively affordable cost of extra devices, thus acting as an intermediate solution. It is foreseeable that with the increasingly wider adoption of consumer-level drones and various sensors over time, this approach will attract more and more attention in both academia and industry due to its real promise of bringing practical MoCap into the daily life of normal people. However, currently, these approaches are still limited in the capture accuracy.

Traditionally MoCap only cares about the body pose without considering other body parts like hair or accessories like handbags and shoes. It is the task of the end-users

to manually add the necessary extra parts to the virtual avatars driven by the obtained body pose or motion. More and more work focuses on providing an end-to-end solution that estimates detailed full-body geometry from monocular images or videos, with or without being textured. Some representative work includes PiFU (Saito et al., 2019), MonoPerfCap Xu et al. (2018b), the method proposed by Alldieck et al. (2018), and more recently the SMPLicit (Corona et al., 2021). There is also work on generative clothed body modeling like CAPE (Ma et al., 2020), SCALE (Ma et al., 2021) and SCANimate (Saito et al., 2021). This thesis mainly focuses on the body pose of the body under clothing itself.

Based on the review of the current mainstream MoCap methods, we believe there is still a strong need for more practical solutions to MoCap. The ideal MoCap system should go beyond pure 3D skeletons and directly provide a 3D mesh. Thus the end-user does not have to manually design an animatalbe template. It also needs to be lightweight and relatively cheap, thus accessible to the public. Even better, the subject should be free to move in an ample space without having to do re-calibration from time to time. MoCap systems with these good properties are what we try to achieve in this thesis.

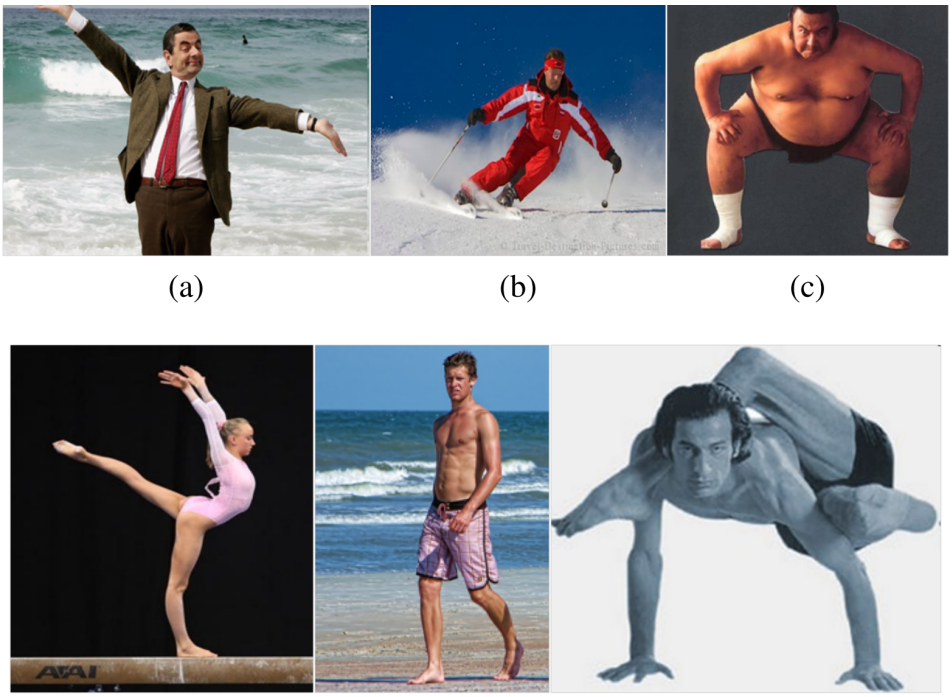
1.2 Challenges

It is intrinsically hard to accurately and efficiently estimate the whole-body configuration from dense range measurement (Dou et al., 2016; Tao et al., 2018; Li et al., 2013; Zhang et al., 2017; Bogo et al., 2014, 2017), let alone to do that from partial or incomplete measurements. The difficulty mainly stems from three factors.

Firstly, the human body is complex to describe, and the spectrum of possible human poses is virtually unlimited, like shown in Figure 1.2. Humans are articulated creatures with many body parts, each able to move freely within certain limits relative to its parent joint. Though most MoCap methods currently only focus on the 3D skeleton, it is a desirable feature for the MoCap system to recover the body surface, one extra property of practical importance to many real applications. How to effectively represent the human body in a realistic fashion remains an open problem. The solution space is enormous even when only 3D joints are considered. It is well known that most Machine Learning algorithms struggle when the dimensionality of the solution space is large.

Secondly, it is challenging to gather a large and diverse dataset covering the most common human poses. Data has proven to be the key for Deep Learning models, whose effectiveness has been validated in many branches of Computer Vision. How to obtain high-quality data remains the first problem for the successful application of Deep Learning in the area of MoCap. Please note that the already proven highly useful crowdsourcing data gathering tools like Amazon Mechanical Turk [5] cannot be directly utilized here since 3D body pose is way harder to annotate compared with 2D keypoints or image cat-

[5] https://www.mturk.com/



(a) (b) (c)

(d) (e) (f)

Figure 1.2: Humans exhibit huge diverse in pose in daily life. This renders accurate tracking of humans hard. Image credit goes to (Pons-Moll, 2014).

Figure 1.3: SMPL is a model that gives both 3d skeleton and body mesh. Shown here is the typical results obtained by fitting SMPL model into monocular image measurement (Bogo et al., 2016a).

egories. This difficulty is partially responsible for the lack of a general and versatile large 3D body dataset. The intrinsic difficulty in assembling standard benchmarks in related areas make the dataset construction work like Human3.6M of (Ionescu et al., 2014b), 3DHP of (Mehta et al., 2017a), FAUST of (Bogo et al., 2014), Dyna of (Pons-Moll et al., 2015) and TOSCA of (Bronstein et al., 2008) valuable and influential. However, these datasets are still limited in diversity and quantity.

Thirdly, humans tend to interact with other objects rather than act independently. Interaction with objects is an indispensable step for humans to execute various tasks. So it is natural and desirable to jointly capture the motion of both the body and objects, rather than treat them separably. However, most previous work only focuses on the human body, ignoring the objects with which the subject interacts. Some methods focus on hands and objects to narrow down the capture space. This assumption renders the problem at hand easier to address, at the cost of realism of human motion. The previous work targeting this problem usually utilizes exotic marker-based systems like the GRAB dataset (Taheri et al., 2020) and the FPHA dataset (Garcia-Hernando et al., 2018). However, this dataset provides no natural RGB images and thus is not directly useable for Computer Vision tasks. The main reason why objects are ignored in most previous work is that it is intrinsically hard to jointly track objects and humans, due to strong occlusions between these.

1.3 Model-based Optimization Methods

For a long time, most mainstream MoCap methods have been based on optimization. These approaches enjoy the features of great applicability, strong reliability and explainability. Typically, three key components are involved: feature extraction, output representation, and loss function minimization, discussed below. This section presents a rough analysis of existing representative work in this category.

The first step in optimization-based MoCap methods is to extract useful features from

input sources that can facilitate further processing. A good feature should be informative of the target, robust to various adversary conditions like dimmer lighting, and computationally economical to obtain. Body contours and 2D keypoints from images or videos (Rhodin et al., 2016b, 2015; Biggs et al., 2020; Zhang et al., 2020a; Joo et al., 2018; Bogo et al., 2016b; Lassner et al., 2017a; Huang et al., 2017), are the most commonly used ones since they compactly encode body shape and pose information. Other common input data modalities include Inertial Measurement Units (IMUs) (von Marcard et al., 2018a,a), electromagnetic signals (Kaufmann et al., 2021), radio frequency signals (Zhao et al., 2019) and pressure sensors (Casas et al., 2019). Still, methods based on dense markers provide the best qualitative and quantitative results so far (Mahmood et al., 2019; Loper et al., 2014).

Another key aspect to consider in developing a MoCap method is the representation format for the human body. A trade-off has to be made between efficiency and effectiveness: a coarse and light representation may make the inference faster and easier, while a more detailed and richer body representation has a better potential of gaining higher quality results. The most basic format is 3D skeletons, the counterpart of 2D keypoints in landmark localization tasks. Blob-like Sum of Gaussians models act as an intermediate representation, while the latest 3D generative body mesh models like SCAPE (Anguelov et al., 2005), GHUM (Xu et al., 2020), SMPL (Loper et al., 2015), SMPL-X (Pavlakos et al., 2019a) and STAR (Osman et al., 2020) push the naturalism and expressiveness to a new level. The recovered meshes exemplified in Figure 1.3 also fit the underlying human quite well.

As for defining and minimizing a loss function, normally, gradient descent methods like LBFGS or conjugate gradient descent are used. The Ceres[6] library is one of the widely used for this purpose. In general, the loss function comprises of two parts, the data terms that measure how the estimated results fit the observations and the regularization terms that serve as priors. These concepts in the context of MoCap are well illustrated in several methods like SMPLify (Bogo et al., 2016a), MonoPerCap (Xu et al., 2017), and Sparse Inertial Poser (Von Marcard et al., 2017). and Sparse Inertial Poser (Von Marcard et al., 2017).

1.4 Learning-based Regression Methods

Learning-based MoCap methods receive more and more attention in both academia and industry after Deep Learning proves its great effectiveness in ImageNet (Deng et al., 2009). Compared with optimization-based alternatives, learning-based algorithms enjoy the advantages of being conceptually simpler, easy to implement, and capable of achieving real-time running speed. Many methods have been proposed to estimate the whole 3D body from images or other sensors. Note that directly regressing body rotations is

[6] http://ceres-solver.org/

Figure 1.4: The DeepCap method proposed in (Habermann et al., 2020) is capable of recovering highly detailed cloth deformations from several normal RGB cameras. Note the estimated fine cloth wrinkles.

much harder than regressing 3D skeletons, a phenomenon observed by many others. This is because, as a collection of point clouds, 3D skeletons reside in a Euclidean space, with a well-defined and continuous metric to optimize, while 3D rotations do not. There are specially designed representation to attenuate this issue like 6D representation (Zhou et al., 2019) and self-selecting Ensembles (Xiang, 2021).

HMR (Kanazawa et al., 2018a) is the first CNN-based MoCap method that directly regresses SMPL pose and shape parameters from one single monocular image. It is very different from previous work where a 3D skeleton acts as the output. They also introduce a novel discriminator to reject unnatural body pose rather than directly adopt the commonly used GMM prior. TexturePose (Pavlakos et al., 2019b) achieves better regression results by making use of texture consistency loss into the pipeline. HoloPose (Guler and Kokkinos, 2019) achieves good results even on in-the-wild images by dividing the output space into discrete clusters, which shows easier for the network to regress compared wihh the continuous counterparts. VIBE (Kocabas et al., 2020) extends HMR to motion sequences and shows smoother reconstruction results. Temporal dependence of natural human motion represented in SMPL format is learned and used as a sequential prior during training. SPIN (Kolotouros et al., 2019) creatively combines both complementary methodologies together to get better results in an iterative way. It is based on the key insight that pseudo ground truth obtained via optimization serves as a stronger signal for network learning. ExPose (Choutas et al., 2020) goes beyond previous methods and also takes face and hands into consideration. It is based on the more recent SMPL-X model (Pavlakos et al., 2019a) where more detailed facial expressions and hand poses are integrated with the base SMPL body model. PIXIE (Feng et al., 2021) and the work proposed

by Zhou (Zhou et al., 2021) consider the correlation relationship between body and face and make use of this information for better whole-body reconstruction. The work proposed in (Jiang et al., 2020) and (Sun et al., 2021) handles multiple persons shown in one single image and estimates all their 3D body configurations in one pass. Recently transformer architextures are also introduced in the area (Lin et al., 2021), and impressive results are yielded. There are also interesting methods that combine physics (Shimada et al., 2020), or reinforcement learning (Peng et al., 2021, 2018) into the pipeline.

All the previous works only consider the naked body itself. This is obviously a huge limitation. After the seminal works represented by ClothCap (Pons-Moll et al., 2017; Zhang et al., 2017; Tiwari et al., 2020) that successfully segments individual clothes from raw scans, statistical 3D clothing models have become more and more mature (Ma et al., 2020, 2021). Correspondingly many works started to recover clothed bodies (Corona et al., 2021; Saito et al., 2019; Alldieck et al., 2018, 2019; Habermann et al., 2020). As demonstrated in Figure 1.4, the current state-of-the-art methods are even able to recover highly fine-grained cloth wrinkle details.

1.5 Contributions

In summary, three main contributions are made in this thesis:

- Firstly, we proposed the first fully-automatic, highly detailed, and generally applicable body mesh estimation method from multiple-view regular RGB videos. Compared with previous methods that generally require manual body template creation and assume coarse body representation, our method directly returns an expressive, subject-specific, and animatable 3D body mesh surface for each frame. The output has the potential to be directly used in various applications with minor or no post-processing. What is more, our method is not limited to laboratory environments and can be relatively easily deployed in the wild. Algorithmically a novel Discrete Cosine Transformation (DCT) prior is integrated into the system to encourage temporally smooth motion estimation. By doing this, the fitting error introduced by the 2D detection error can also be corrected.

- Secondly, we explore the possibility of jointly tracking the whole-body motion of the subject and the motion of the object the subject is interacting with, from a set of six RGB-D Kinect cameras. We first do motion estimation for the subject and the object separately by fitting the parametric body model and 3D mesh of the object into 2D contours and keypoints obtained via state-of-the-art neural networks. Then we refine the results in one go by taking smoothness and physical constraints into consideration. We obtain reasonable results for ten everyday objects, like a suitcase and an umbrella, and gathered a novel 3D body and object interaction dataset with RGB footage.

- Lastly, we address the problem of practical and real-time body pose estimation from lightweight hardware devices. We show that it is feasible to obtain decent body tracking results from as few as six sparse IMUs at an interactive rate. Our method is the first to directly do regression on 3D joint rotations in an inpainting way. We proved that a model trained on a synthetic dataset generalizes well on unseen real data in our setting. We also tried estimating what the subject is doing from a single wrist-worn smartwatch. Our simple yet effective method achieves an average accuracy of around 80% for Human Activity Recognition (HAR). We hope our study can also stimulate more thinking about the privacy issues related to the intelligent devices common in daily life nowadays.

1.6 Thesis Organization

The thesis is structured as follows.

- Chapter 1. **Introduction**: A rough overview of the problem to target, existing solutions, pros and cons of these methods, and the contributions made in this thesis.
- Chapter 2. **Foundations**: Description of the main mathematical, software, and hardware tools used in all works presented in this thesis, including the statistical 3D generative body models, differentiable renderers, shape and pose priors learned from data, and IMUs.
- Chapter 3. **Related Work**: Summary of previously published works that are related to the works presented in this thesis. Through this discussion, the works here are positioned in the global picture.
- Chapter 4. **Multi-view SMPLify**: An algorithm to jointly estimate the shape and pose of the subject over time from multiple spatially and temporally calibrated RGB cameras. A generative 3D body model like SMPL is assumed. This method works in two consecutive stages: frame-wise shape and pose estimation and joint whole-sequence refinement of the motion across time. In the end, one accurate, personalized body mesh and temporally stable motion represented as meshes are obtained. The output meshes are animation-ready and directly usable for many applications. This chapter is based on the work of Huang et al. (2017).
- Chapter 5. **Joint Human and Object Tracking**: An endeavour to extend body-only MoCap with the ability to track rigid objects the subject interacts with. The pre-defined rigid objects are scanned first to obtain their geometries represented as meshes. The subject and the object are firstly tracked separately by fitting the articulated body model, and the rigid object meshes into 2D measurements. Then the interaction and contact constraints between

these two are considered together in an integrated optimization. This chapter is based on the work of Huang et al. (2022).

- Chapter 6. **Real-time MoCap from Sparse IMUs**: Presented in this chapter is the first system to do whole-body motion capture from only six IMUs. In comparison, mainstream IMUs-based MoCap suits require 17 or more IMUs. This system is based on LSTM and runs in real-time. Tracking performance is on par with, or even better than, optimization-based alternative algorithms on standard benchmarks. The idea of firstly training the model on large-scale synthetic data and then fine-tuning it on real data is presented and proven effective in this work. A pipeline to turn archival MoCap marker data into synthetic IMU readings paired with SMPL ground truth poses is introduced. This chapter is based on the work of Huang et al. (2018).
- Chapter 7. **Activity Estimation from One Smartwatch**: In this chapter, Human Activity Recognition from one smartwatch is described. The main motivation is to investigate what useful information about humans can be obtained from minimal electronic equipment. It is shown that the everyday activity the subject is doing can be accurately inferred from a single IMU inside a consumer-level smartwatch. This finding also reveals the real danger of getting one's privacy (like what he/she tends to do at a specific time) leaked via smart devices that are usually reckoned to be safe. The work presented in this chapter has not been published.
- Chapter 8. **Conclusion and Future Work**: At the end is an overall summary of the whole thesis, followed by discussions about possible future work along the direction of MoCap.

Chapter 2

Foundations

In this chapter, we review the major supporting technologies used in this thesis. The core to the work done in this thesis includes a sophisticated generative 3D Human Body Model (SMPL Loper et al. (2015) and SMPL-X Pavlakos et al. (2019a)) that serves as the output, an approximately auto-differentiable renderer (OpenDR Loper and Black (2014)) that enables the analysis-by-thesis framework, a method (MoSh Loper et al. (2014)) to turn archival optical marker data into compatible SMPL format, advanced methods to extract semantic information about humans from images, and the effective Deep Learning methods that make learning from large datasets possible.

2.1 SMPL Model and its Variants

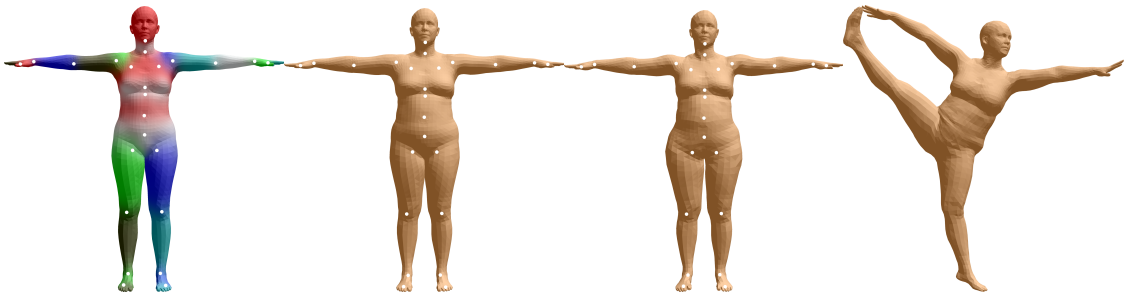
SMPL Loper et al. (2015) is a parametric model of 3D human body shape and pose that takes 72-d pose, and 10-d or higher dimensional shape, parameters, θ and β respectively, and returns a mesh with $N = 6890$ vertices. Shape and pose deformations are applied to a base template, T_μ, that corresponds to the mean shape of training 3D scans. Mathematically the mapping from the parameters to the final body surface vertices is represented as:

$$M(\beta,\theta) = W(T(\beta,\theta),J(\beta),\theta,\mathbf{W}) \tag{2.1}$$

$$T(\beta,\theta) = T_\mu + B_s(\beta) + B_p(\theta), \tag{2.2}$$

where W is a linear blend-skinning (LBS) function applied to the template mesh in the rest pose, to which pose- and shape-dependent deformations, $B_p(\theta)$ and $B_s(\theta)$, are added. The resulting mesh is then posed using LBS with rotations about the joints, $J(\beta)$, which depend on body shape. The shape-dependent deformations model subject identity while the pose-dependent ones correct LBS artifacts and capture deformations of the body with pose. The whole pipeline of SMPL animation for a specific pose is shown in Figure 2.1.

SMPL has many advantages over the predecessor SCAPE Anguelov et al. (2005). Firstly SMPL is conceptually much simpler to understand, totally linear, and can be computed in real-time. It is also compatible with existing graphics engines. These features



(a) Template shape (b) + shape-blend-shape (c) + pose-blend-shape (d) Final animated mesh

Figure 2.1: Illustration of how SMPL works. Starting with the mean body shape in canonical T-pose, firstly the shape variation due to different identities (named shape-blend-shape), are added, then the shape variation due to different poses are added (called pose-blend-shape), finally the animated mesh is obtained via LBS.

make it the first choice for 3D human-related applications. Secondly, SMPL connects body joints with surface vertices in a linear, thus differentiable way. This design permits estimating 3D body shape and pose from sparse joint estimates, as evidenced in SMPLify Bogo et al. (2016b). Thirdly SMPL is learned from a much larger dataset via the state-of-the-art Co-registration framework Hirshberg et al. (2012). During training, the SMPL model and scan registrations are updated iteratively. In the end, a better model can be expected.

SMPL is intrinsically limited in representing the face and hands, the relatively small and independent subparts of the human body, because it is a model initially designed for the whole body. To address this issue FLAME Li et al. (2017) and MANO Romero et al. (2017) are proposed. To be compatible with SMPL, FLAME and MANO share the same design philosophy as SMPL: mean shape template serving as the starting point for 3D surface, PCA components representing shape variation in canonical pose, and pose-blend-shapes attenuating variation artifacts caused by LBS. The combination of all these three individual components results in a complete and fully functional model called SMPL-X Pavlakos et al. (2019a). More recent versions enforce sparsity constraint for the blending weights Osman et al. (2020) and also add foot joints Osman et al. (2022).

2.2 MoSh and OpenDR

Having been the de-facto industrial standard for quite a long time, optical marker-based MoCap like Vicon has produced a large amount of data distributed at different places. Though covering diverse types of natural human motion, these archival data can be dramatically different in the number of optical markers and the placement positions, thus not directly usable for machine learning algorithms. MoSh Loper et al. (2014) is the method that can convert all these data into the same format. MoSh works by deforming

the generative human body model to match the measurements of optical markers. It supports any number of optical markers and the sliding of markers. It can not only recover accurate body shape and pose, but also yields realistic soft tissue deformations. One application of MoSh is the popular AMASS Mahmood et al. (2019) dataset, which consists of a lot publicly available datasets that are processed by MoSh. AMASS is currently the largest 3D human dataset that is in SMPL-X format. It can be used for learning human pose priors like VPoser Pavlakos et al. (2019a) or human motion models.

A renderer is an algorithm that turns the 3D representation of objects or scenes into 2D images. In some sense, Computer Vision can be treated as "inverse graphics" Baumgart (1974), where the input are 2D measurements, and the output is the corresponding 3D counterparts of interest. OpenDR Loper and Black (2014) is the first practical auto-differentiable renderer that enables easy-to-use 3D body shape and pose estimation from 2D image data. OpenDR is based on Chumpy [1] and OpenGL [2]. It supports optimization of vertices, texture, and lighting. Later there are various follow-up works Kato et al. (2018); Genova et al. (2018); Liu et al. (2019); Ravi et al. (2020) implemented in Tensorflow Abadi et al. (2016b) and PyTorch Paszke et al. (2019) to better integrate with Deep Learning models.

2.3 2D Keypoints and Silhouettes

For most Machine Learning algorithms, the loss function is one critical component. In the framework of analysis-by-synthesis, it encodes the metric of how well the synthesis matches the measurement and the quality of reconstructions. In work presented in this thesis we mainly consider four types of information about humans: the 2D keypoints, the body contour, limb orientation and acceleration, and 3D point clouds. Of them, the 2D keypoints and silhouettes are extracted from RGB videos via Deep Learning algorithms.

Arguably the most popular 2D body keypoint localization algorithm is OpenPose Cao et al. (2019), though there are other great alternatives like AlphaPose Fang et al. (2017); Li et al. (2018); Xiu et al. (2018). Initially implemented in Caffe Jia et al. (2014), OpenPose supports the detection of whole body key landmarks, including those of the face and hands Simon et al. (2017). Given one RGB image containing human bodies, OpenPose yields the 2D location prediction of the major body keypoints and the prediction confidence. The confidence value can be naturally used as the weighting in the loss terms.

For the human segmentation method, we initially adopted the same one used in Unite-the-People Lassner et al. (2017a), then we switched to the more recent DeepLab implementation Chen et al. (2017b). We empirically found that these methods work well for most cases. Imperfect predictions only happen when the background is cluttered or vi-

[1] https://github.com/mattloper/chumpy
[2] https://www.opengl.org/

sually hard to distinguish from the subject. We also tried Mask-RCNN He et al. (2017) and mainly used it for object segmentation and detection.

2.4 Priors for Pose and Motion

This thesis focuses on the recovery of whole-body pose and motion from different data modalities that provide partial and incomplete information. This problem is intrinsically ill-posed since countless possible solutions can generate the same 2D input. So to encourage the prediction to be natural and human-like, we need to apply priors to body shape and pose. This section discusses various priors for body pose since the plausible pose space is large and harder to model.

The most intuitive way to model a pose prior is via Gaussian Mixture Models (GMM) on the axis-angle representation of body pose as used in SMPLify Bogo et al. (2016b). The pose-conditioned joint limit prior is also explored in Akhter and Black (2015) However, GMM presumes that the space to model can be well approximated by a linear combination of Gaussian distributions, which might be a too strong assumption for natural body pose. Lately, many different priors are proposed to address this issue, like the discriminative prior used in HMR Kanazawa et al. (2018b), VPoser in SMPLify-X Pavlakos et al. (2019a), and Normalizing flows Zanfir et al. (2020); Biggs et al. (2020).

The next is to model natural human motion over a sequence. There has been many works focusing on predicting human motion in the future from past observations Ghosh et al. (2017); Aksan et al. (2019); Martinez et al. (2017); Barsoum et al. (2018); Aksan et al. (2019). The sequential discriminator proposed in VIBE Kocabas et al. (2020) suits this purpose quite well. In contrast to these data-driven models, the Discrete Cosine Transform Jain (1989) analytically provides a set of basis vectors representing cyclic motion from low frequency to high frequency. This feature of DCT has been successfully applied to model various natural events in Akhter et al. (2010).

2.5 Inertial Measurement Units

Inertial Measurement Units (IMUs) are electronic sensors that can measure the orientation and acceleration of the object they are attached to. IMUs fit perfectly the problem of MoCap since they are not affected by occlusion, are lightweight to carry, and directly give information closely related to body pose. There are even commercial solutions Roetenberg et al. (2009) for MoCap which purely rely on IMUs.

A fully functional IMU-based MoCap system needs 17 IMUs to cover all major human body joints that are free to move. The configuration adopted in XSens is shown in Figure 2.2. Please note that by default, the IMU measurements are relative to the inertial frame, while for various human body models, the local frame of each joint is defined relative to its direct parent in the kinematic tree. So a step of calibration is required before the data

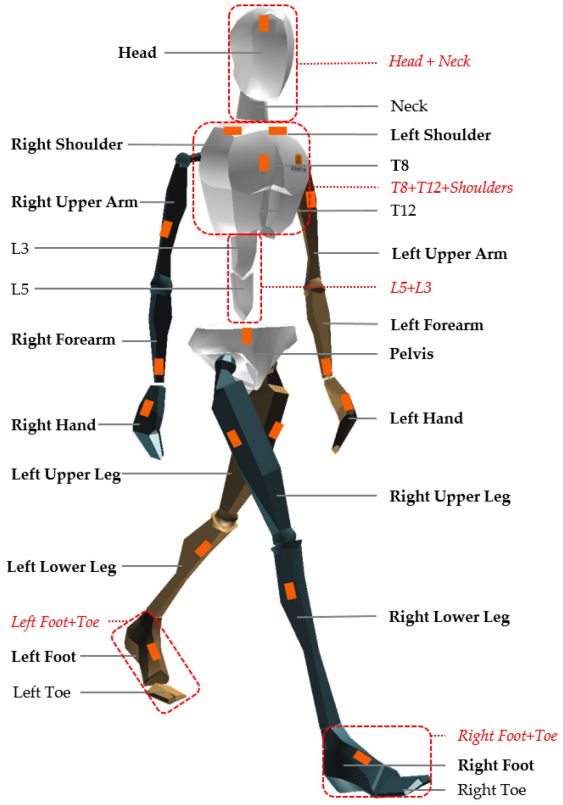


Figure 2.2: The placement of 17 IMUs in the XSens MoCap suit.

capture. We will talk more about the calibration procedure later.

2.6 Dataset

Depending on how the data is captured, datasets in the field of (marker-less) Motion Capture can be roughly divided into the ones captured inside or laboratory environments (indoor) or the ones captured in the wild. One difficulty in constructing benchmarks in this field is that the acquisition of 3D ground-truth is far from trivial.

HumanEva Sigal et al. (2010) and Human3.6M Ionescu et al. (2013) are the standard benchmarks to train, test and compare different algorithms. They are both captured inside laboratory environments, with optical markers placed on the subjects for ground-truth 3D pose construction. HumanEva is relatively small, containing four subjects performing six different motions in 56 sequences. Human3.6M is larger, with 11 subjects performing 15 types of activities in around 3 million frames. Both datasets provide multi-view RGB images and the corresponding 3D ground-truth skeletons.

TNT15 Von Marcard et al. (2016) and TotalCapture Trumble et al. (2017) are another two indoor datasets that also contain IMU measurements. The ground-truth 3D pose

of TNT15 is obtained from 10 IMUs, while for TotalCapture, the ground-truth is provided by the Vicon system. The first large-scale and in-the-wild dataset titled 3DPW is proposed in von Marcard et al. (2018b). There the image data together with IMU measurements are jointly utilized to fit the 3D body model SMPL. Unlike previous methods that ignore the possible sliding issue of the IMU measurements, the algorithm proposed in von Marcard et al. (2018b) propose to tackle this problem by explicitly estimating the measurement gap. It is shown in the paper that the reconstruction error is around 26mm, which is accurate enough to serve as a benchmark.

Still, in general it is quite hard to gather high-quality in the wild data for 3D body pose estimation. In contrast, researchers have also proposed to generate synthetic 2D measurements from 3D body meshes. The representative work includes Deep3DPose Chen et al. (2016) and SURREAL Varol et al. (2017). The pros and cons of these methods complement the ones discussed before: the 3D ground truth is exact while the 2D rendering might not be so realistic.

Chapter 3

Related Work

3.1 Optimization-Based Methods from Cameras

Marker-less MoCap methods are becoming more and more popular for their wide application scenarios and easy setup. Most previous works focus on one aspect of the two closely related problems: markerless 3D human body shape and pose estimation. Some of these target 3D pose estimation (Amin et al., 2013; Deutscher and Reid, 2005; Du et al., 2016; Gall et al., 2010; Ramakrishna et al., 2012; Sigal et al., 2012; Yao et al., 2011). They formulate it as a discriminative problem, directly inferring 3D pose from 2D image features, assuming no explicit human body model. Amin et al. (Amin et al., 2013) extend single-view based pictorial structures to multi-view cases, jointly infer the 2D joint location of all views, then use linear-triangulation to obtain the 3D joints. Yao et al. (Yao et al., 2011) propose a stochastic gradient-based method for a Gaussian Process Latent Variable Model (GPLVM), which shows good optimization properties. Uncertainty over estimated 2D image features has also been considered. Zhou et al. (Zhou et al., 2016) introduce a sparsity prior over human pose and jointly handle the pose and 2D location uncertainty, while Kazemi et al. (Kazemi et al., 2013) address the body part correspondence problem by optimizing latent variables. Similar ideas are proposed by Simo-Serra et al. (Simo-Serra et al., 2013), in which they also estimate 2D and 3D pose at the same time. Twin Gaussian processes (Bo and Sminchisescu, 2010) have also been used on this problem. Recently deep learning methods achieved the most accurate pose estimation results (Du et al., 2016; Moreno-Noguer, 2016; Popa et al., 2017; Tekin et al., 2015; Trumble et al., 2016). To address their need for a massive amount of training data, Yasin et al. (Yasin et al., 2016) propose a dual-source approach. Pavlakos et al. (Pavlakos et al., 2017a) directly regress 3D pose from RGB image via CNNs in a coarse-to-fine manner.

The second primary set of approaches use an explicit intermediate human body representation, which effectively assists pose estimation but often lacks realism (Belagiannis et al., 2014; Deutscher et al., 2000; Sigal et al., 2012; Stoll et al., 2011). Common human body representations include the Articulated Human Body Model (Deutscher and Reid, 2005), 3D Pictorial Structures (Belagiannis et al., 2014; Sigal et al., 2012), the sum-of-Gaussians model (Rhodin et al., 2015; Stoll et al., 2011), and the Triangulated Mesh

Model (Sigal et al., 2007). These models are usually utilized to represent the structure of the human body, thus facilitating the inference of pose parameters. Sometimes the body mesh is also considered, but in an abstract or coarse way, without considering the shape details.

Estimating both the pose and surface mesh usually requires complex global optimization (Gall et al., 2010, 2009). Often the silhouette of the body is assumed to be known (Bălan and Black, 2008) and manual initialization or a pre-scanned surface mesh is required (Ahmed et al., 2005; Ballan and Cortelazzo, 2008; De Aguiar et al., 2008; Hasler et al., 2010; Ilic and Fua, 2006; Jain et al., 2010; Plankers and Fua, 2003; Starck and Hilton, 2003; Wu et al., 2012; Vlasic et al., 2008). Balan et al. (Balan et al., 2007) address this problem by fitting a SCAPE body model (Anguelov et al., 2005) to multi-view silhouettes. Their initialization method is complex, and they do not integrate information over time. Another very recent work, concurrent with ours, is the one proposed in (Pavlakos et al., 2017b). They also use CNNs to detect 2D joints, then fit a 3D pictorial structures model to the detections. Their method only returns 3D joints as output, while ours estimates body shape and pose together. The method proposed in (Mehta et al., 2017b) simultaneously regresses 2D and 3D joints from monocular video via one CNN, then fits a skeleton model to the 3D joint estimations, achieving real-time performance. The recent series of work Habermann et al. (2019, 2021b, 2020) make it possible to accurately reconstruct personalized 3D body surfaces from monocular or multi-view videos in real-time, including even highly detailed and intricate cloth deformations.

The most similar recent work addresses the fully automatic estimation of 3D pose and shape from monocular images (Bogo et al., 2016b; Xu et al., 2018b; Shimada et al., 2022; Rempe et al., 2021; Yi et al., 2022), and multi-views videos (Rhodin et al., 2016b; Zhang et al., 2021b; Zheng et al., 2021; Zhang et al., 2020b). The SMPLify algorithm proposed by Bogo et al. (Bogo et al., 2016b) makes it possible to simultaneously obtain a 3D pose and convincing body shape from a single image without requiring any user intervention and without assuming background extraction or complex optimization techniques. Based on the state-of-the-art 3D human body model, SMPL (Loper et al., 2015), they infer the human shape and pose parameters by fitting the projection of 3D SMPL joints to 2D joints estimated via a 2D joint detector like DeepCut or CPM (Pishchulin et al., 2016; Wei et al., 2016). Ambiguity issues are handled by applying priors learned from the large-scale public CMU dataset (cmu, 2000), which is vital for their method to yield valid results. Rhodin et al. (Rhodin et al., 2016b) propose a method that works on multi-view videos. Built upon a sum-of-Gaussian shape model (Rhodin et al., 2015; Stoll et al., 2011), their algorithm first initializes the pose of each Gaussian blob, then refines the pose and shape of each blob via the body contour approximation with image gradients. As in Bogo et al. (Bogo et al., 2016b), they use deep learning to estimate 2D joints to get rough joint locations in each view. They enforce temporal coherence by penalizing acceleration between frames.

3.2 Optimization-Based Sensor Fusion Methods

Inertial trackers Commercial inertial tracking solutions (Roetenberg et al., 2007) use 17 IMUs equipped with 3D accelerometers, gyroscopes and magnetometers, fused together using a Kalman Filter. Assuming the measurements are noise-free and contain no drift, the 17 IMU orientations completely define the full pose of the subject (using standard skeletal models). However, 17 IMUs are very intrusive for the subject, long setup times are required, and errors such as placing a sensor on the wrong limb are common. To compensate for IMU drift, the pioneering work proposed in (Vlasic et al., 2007) uses a custom system with 18 boards equipped with acoustic distance sensors and IMUs. However, the system is also very intrusive and difficult to reproduce.

Video-inertial trackers Sparse IMUs have also been combined with video input (Pons-Moll et al., 2010, 2011; von Marcard et al., 2016; Malleson et al., 2017), or with sparse optical markers (Andrews et al., 2016) to constrain the problem. Similarly, sparse IMUs have been combined with a depth camera (Helten et al., 2013); IMUs are only used to query similar poses in a database, which constrain the depth-based body tracker. While powerful, hybrid approaches that use video suffer from the same drawbacks as pure camera-based methods, including occlusions and restricted recording volumes. Recent work uses a single moving camera and IMUs to estimate the 3D pose of multiple people in natural scenes (von Marcard et al., 2018a), but the approach requires a camera that follows the subjects around.

Optimization from sparse IMUs Von Marcard et al. (Von Marcard et al., 2017) compute accurate 3D poses using only 6 IMUs. They take a generative approach and place synthetic IMUs on the SMPL body model (Loper et al., 2015). They solve for the sequence of SMPL poses that produce synthetic IMU measurements that best match the observed sequence of real measurements by optimizing over the entire sequence. Like (Von Marcard et al., 2017) we also use 6 IMUs to recover full-body pose, and we also leverage SMPL. However, our approach is conceptually very different: instead of relying on computationally expensive offline optimization, we learn a direct mapping from sensor data to the full pose of SMPL, resulting in real-time performance and good accuracy despite using only 6 IMUs.

3.3 Learning-Based Methods

Sparse accelerometers and markers An alternative to sensor fusion and optimization is to learn the mapping from sensors to full body pose. The human pose is reconstructed from 5 accelerometers by retrieving pre-recorded poses with similar accelerations from a database (Slyper and Hodgins, 2008; Tautges et al., 2011). The mapping from acceleration alone to the position is, however, very difficult to learn, and the signals are typically

very noisy. A somewhat easier problem is to predict a full 3D pose from a sparse set of markers (Chai and Hodgins, 2005); here, online local PCA models are built from the sparse marker locations to query a database of human poses. Good results are obtained since 5-10 marker positions constrain the pose significantly; furthermore, the mapping from 3D locations to pose is more direct than from accelerations. This approach requires a multi-camera studio to capture reflective markers.

Motion sensors Alternatively, position and orientation can be obtained from motion sensors based on inertial-ultrasonic technology. Full pose can be regressed from 6 such sensors (Liu et al., 2011), which provide global orientation and position. While global position sensors greatly simplify the inverse problem since measurements are always relative to a static base station; consequently, capture is restricted to a pre-determined recording volume. Furthermore, such sensors rely on a hybrid inertial-ultrasonic technology, which is mostly used for specialized military applications.[1] Our method uses only commercially available IMUs —providing orientation and acceleration but no position, and does not require a fixed base-station.

Sparse IMUs Learning methods using sparse IMUs as input have also been proposed (Schwarz et al., 2009), where full pose is regressed using Gaussian Processes. The models are trained on specific movements of individual users for each activity of interest, which greatly limits their applicability. Furthermore, Gaussian Processes scale poorly with the number of training samples. Generalization to new subjects and un-constrained motion patterns is not demonstrated.

Locomotion and gait IMUs are often used for gait analysis and activity recognition, recently combined with deep learning approaches (cf. (Wang et al., 2017), for example, to extract gait parameters via a CNN for medical purposes (Hannink et al., 2016). Prediction of locomotion has been shown using a single IMU (Mousas, 2017) using a hierarchical Hidden Markov Model. Deep learning has been used to produce locomotion of avatars that adapt to irregular terrain (Holden et al., 2017), or that avoid obstacles and follow a trajectory (Peng et al., 2017). These approaches are suited to cyclic motion patterns, where the cycle phase plays a central role.

In summary, existing learning methods either rely on global joint positions as input—which requires external cameras or specialized technology—or are restricted to predefined motion patterns. Thus it is desired to have an algorithm that is based on learning, thus have the potential of running in real-time, while at the same time requiring as less hardware devices as possible.

[1] http://www.intersense.com/pages/20/14

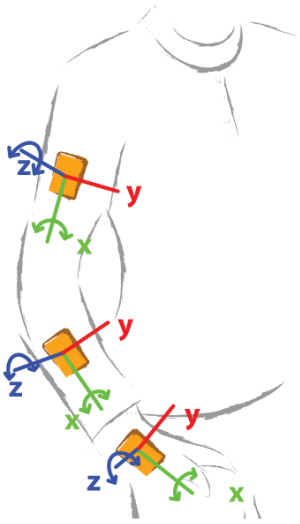
3.4 Human-Object/Scene Interaction

There has been a lot of work targeting modeling or analyzing human and object interactions (Yao and Fei-Fei, 2010; Oikonomidis et al., 2011; Tzionas et al., 2016; Hampali et al., 2020; Hasson et al., 2019; Karunratanakul et al., 2020). A detailed discussion of these methods is out of the scope of this work. Here in this work, we focus on modeling and analyzing human-object interaction in 3D space. Previous works like HONNotate (Hampali et al., 2020) only focus on hand and object motion, ignoring the strong relation between body motion, hand motion, and object motion. Unlike these approaches, we propose to jointly track the body and object together and estimate a reasonable hand pose whenever possible.

Motivated by the observation that natural human motions always happen inside 3D scenes, researchers have proposed to jointly model human motions, and the surrounding environments (Hassan et al., 2019; Savva et al., 2016; Cao et al., 2020). In PROX, the contact between humans and scenes is explicitly used to resolve the ambiguity problem in pose estimation. The avoidance of human-scene penetration is also enforced in the model fitting. People also try to infer the most plausible position and pose of the humans given the 3D scenes (Li et al., 2019). The prediction of long-term motion conditioned on the 3D scenes is also explored. Unlike these methods going from 3D scenes to human pose and motion, PiGraph (Savva et al., 2016) learns the most common interaction pattern between humans and scenes.

3.5 HAR from IMUs

IMU-based Human Activity Recognition (HAR) has attracted much attention for its appealing features of being highly flexible, lightweight, and capable of working for a long time, thus very suitable for health-care and clinical applications (Attal et al., 2015; Bao and Intille, 2004; Jordao et al., 2018; Bulling et al., 2014; Aggarwal and Ryoo, 2011; Bachlin et al., 2009; Chen et al., 2006; Lester et al., 2006). Usually, it is formulated as a classification task, with the IMU readings as input and the corresponding activity labels as output. Most of the previous works assume a fixed and monotonous scenario. Correspondingly the activities of consideration are limited in diversity. Depending on the specific experimental setting, the ground-truth label of the motion can either be generated readily when the dataset is constructed or manually labeled in a separate stage later on. The places that have been heavily considered include kitchens, factories, hospitals, working offices, and the corresponding activities are site-specific such as Open door (Chavarriaga et al., 2013), Eat (Amft et al., 2005), Run (Reiss and Stricker, 2012) and Pick order (Grzeszick et al., 2017), or even smoking (Añazco et al., 2018). Please note that these works usually assume that more than one industrial IMUs are available (Maurer et al., 2006). The typical number ranges from 3 to 12. The IMUs are placed on the actor's different body parts, hoping to infer as much motion information as possible.



Activities

opening a window
closing a window
watering a plant
turning book pages

drinking from a bottle
cutting with a knife
chopping with a knife
stirring in a bowl

forehand
backhand
and smash

Figure 3.1: Typical IMU placement and activity categories chosen for IMUs-based HAR borrowed from (Bulling et al., 2014). Note that multiple (3 here) commercial IMUs are strapped at different body parts, and all the activities to classify tend to happen inside a kitchen, thus not so representative of the normal human activities. Instead, our system assumes only one IMU-equipped smartwatch and considers 15 common daily activities for different scenarios.

One representative setup is illustrated in Figure 3.1.

Many kinds of methodologies have been applied to this problem. The most commonly adopted Machine Learning algorithms used to be K-Nearest Neighbors (KNN), Hidden Markov Models (HMM) (Ordonez et al., 2014; Blanke and Schiele, 2009) and Random Forests (RF) (Bonomi et al., 2009). Usually, these algorithms work on carefully hand-crafted features extracted from the raw input signals via a pre-processing stage. As normally done in other signal processing areas, another common practice is turning the signal into a frequency representation through the Fast Fourier Transformation (FFT), then combining the FFT output and raw signals. This hybrid data organization tends to yield superior performance for the complementary nature of the temporal domain and frequency domain. These staged systems enjoy clear structures, great interpretability, and stable performance. Contrary to these classic methods, nowadays, more and more people are turning to Deep Learning (Yao et al., 2018), inspired by its huge success in Computer Vision (LeCun et al., 2015). These Deep Learning models (Rippel et al., 2015) try to learn the mapping from raw signal inputs to the corresponding class labels in an end-to-end way. No other manual labor is required except the design of network structure and fine-tuning of hyper-parameters like learning rate. For these systems, it is generally hard to tell which part is responsible for feature extraction and which part is handling classification since the whole system is seamlessly coupled together. Convolutional Neural Networks (Moya Rueda et al., 2018; Zeng et al., 2014; Ronao and Cho, 2015; Yang et al., 2015; Hammerla et al., 2016), Long-Short Term Memory (LSTM) (Valarezo et al., 2017; Chen et al., 2017a; Baldominos et al., 2018; Ordóñez and Roggen, 2016) are among the variants that receive the most attention. Other kinds of wearable sensors have also been integrated in IMUs-based systems for better performance like air-pressure modules (Yang et al., 2018), and mobile phones (Bayat et al., 2014; Ronao and Cho, 2015; Ravi et al., 2016). Rather than focusing on whole-body motions, some works address hand activity or arm activity (Amft et al., 2005).

Though promising, these systems mentioned above, where multiple commercial IMUs are involved, suffer from the disadvantages of being expensive and relatively intrusive, thus not so practical in real life. On the other hand, nowadays, IMU-equipped mobile phones and smartwatches are becoming increasingly ubiquitous and user-friendly. Besides their typical functionalities like texting and making a phone call, these devices also provide a well-supported software development environment, permitting customized intelligent APPs. Some researchers have proposed to do HAR via mobile phones and/or smartwatches (Yang et al., 2018; Laput and Harrison, 2019; Bayat et al., 2014), to truly make the HAR system practical in daily life and full time. Our system only needs a normal Appple smartwatch for information extraction. We develop a WatchOS APP that runs on the smartwatch and keeps checking the IMU measurement. The interface of the APP is displayed in Figure 7.1.

Most closely related to our work is the one recently published in (Laput and Harrison, 2019). Similar to our settings, they also propose to do automatic HAR from one single wrist-worn smartwatch. They gather a large-scale manually labeled IMU dataset, then

train a Neural Network to do classification. But their work mainly focuses on fine-grained hand activities rather than whole-body motions, the core of our study. Also, they record each hand activity instance individually, without considering and modeling the temporal context of natural motions, while in our study, we try to capture and address the natural, continuous, and realistic motions. We see this work as the natural enhancement and expansion of the state-of-the-art. We believe our experimental setting is closer to reality, and we further demonstrate the importance of direct temporal modeling at the algorithmic level.

3.6 Privacy Issues with Smart Devices

Our study is also motivated by the privacy and security issues the whole society faces in the era of mobile and intelligent computing. There is a long history of the public worrying about what side effects new technologies can bring to us (Agre and Rotenberg, 1998; Rowe, 2014; Krishnan, 2016; Ford, 2015). Nowadays, with the enormous popularity of AI and DL, quite a considerable portion of this turns into concern about the violation of personal privacy and abuse of personal data (Abadi et al., 2016a; Barrat, 2013; Papernot et al., 2016; Agarwal et al., 2019). The threats are mainly two-fold: generally, AI or ML models consume a large quantity of real data to learn the underlying patterns, and it is not so easy for people to keep track of whether their personal information has been gathered for this kind of purposes. The unprecedented realism shown in facial or human images generated via GANs (Nagano et al., 2018; Thies et al., 2016; Chan et al., 2018; Kim et al., 2018) illustrates how potentially vulnerable everyone is to these technologies.

Accompanying the rise of intelligent algorithms and systems are the widespread usage and growing functionality of all kinds of intelligent personal devices, represented by intelligent speakers, mobile phones, and smartwatches. They are cheap enough to be generally affordable and powerful enough to support everyday computing needs. Usually, they are equipped with many kinds of gadgets such as IMUs, GPS sensors, and infrared sensors. With the help of these sensors, they can provide many more services beyond traditional ones like making a phone call and texting. Many of these applications are driven by the equipped IMUs and concentrated on the clinical and health care sectors. The everyday use cases are automatic step counting, heart rate monitoring, sleep time tracking, and body stress estimation. Please note that almost all of these applications mentioned above are based on heuristic rules, with little or no learning involved. The measured or inferred information is generally fine-grained, lacking semantic meaning or high-level intention. Take step counting as one example. A straightforward way is to set a threshold for linear acceleration or angular rate. Anytime the detected IMU readings are above this threshold, the counter of the steps increases by one. The threshold can be adjusted for the specific user to make it more flexible.

There has been some work trying to infer human-level behavior understanding by using one single smartwatch or mobile phone. It has been proved that advanced machine

learning techniques can achieve reasonable human activity recognition in specific scenarios. Our work goes one step further along this line of research. We carefully choose 15 everyday daily activities that happen in different places like the kitchen and supermarket and record the human motions in an unconstrained and natural way. Our setting is different from previous works where only one type of activity is recorded in each session by design. Though data captured in that way is easy to label, we found the temporal coherence and transition patterns of human activities are totally lost. We empirically prove that it is of vital value to encode these features in the modeling and inference stage.

By doing so, together with the previous work, we manifest the possible severe privacy threats introduced by these smart devices. As human beings, what we say and what we do largely reflect who we are. Many personal habits and preferences can be inferred by knowing what activities the person is doing most of the time. Activities are closely related to places. Thus it is also possible to decide where the target has been at a specific time segment. More consideration and actions need to be paid to this issue to keep us away from potential leakage and abuse of personal privacy information.

Chapter 4

Multi-view SMPLify

4.1 Introduction

The markerless capture of human motion (mocap) has been a long-term goal of the community. While there have been many proposed approaches and commercial ventures, existing methods typically operate under restricted environments. Most commonly, such methods exploit background "subtraction," assuming a known and static background, and the most accurate methods employ strong prior assumptions about the actor's motion. In many cases, the best results on benchmarks like HumanEva (Sigal et al., 2010) are obtained by training on the same motion by the same actor as is evaluated at test time (Amin et al., 2013). Here we provide a solution for markerless mocap that is more accurate than the current state of the art but is also less restrictive[1].

There are four critical components to our approach. First, our approach exploits SMPL (Loper et al., 2015), a realistic, low-dimensional, 3D parametric model of the human body. Second, we use a Convolutional Neural Network (CNN) to compute putative 2D joint locations in multiple camera images. We then fit the 3D parametric model to the 2D joints robustly. This extends the SMPLify approach for pose and shape estimation (Bogo et al., 2016b) from a single image to multi-camera data.

Third, we go beyond SMPLify (Bogo et al., 2016b) to use a deep CNN to segment people from images (Lassner et al., 2017b). This removes the need for a background image and makes the approach more general. We fit our 3D body model to both the 2D joints and the estimated silhouettes and show that the silhouettes significantly improve the accuracy and realism of the mocap.

Since 2D joints estimated by CNNs sometimes confuse left and right parts of the body, the image evidence alone is not enough for a reliable 3D solution. Consequently, we exploit temporal information to resolve such errors. This leads to the fourth component in which we exploit a generic temporal prior based on the insight that human motions can be captured by a low-dimensional Discrete Cosine Transform (DCT) basis (Akhter et al., 2012). We implement this DCT temporal term robustly and show that it improves performance yet requires no training data.

[1] This chapter is based on the work of Huang et al. (2017).

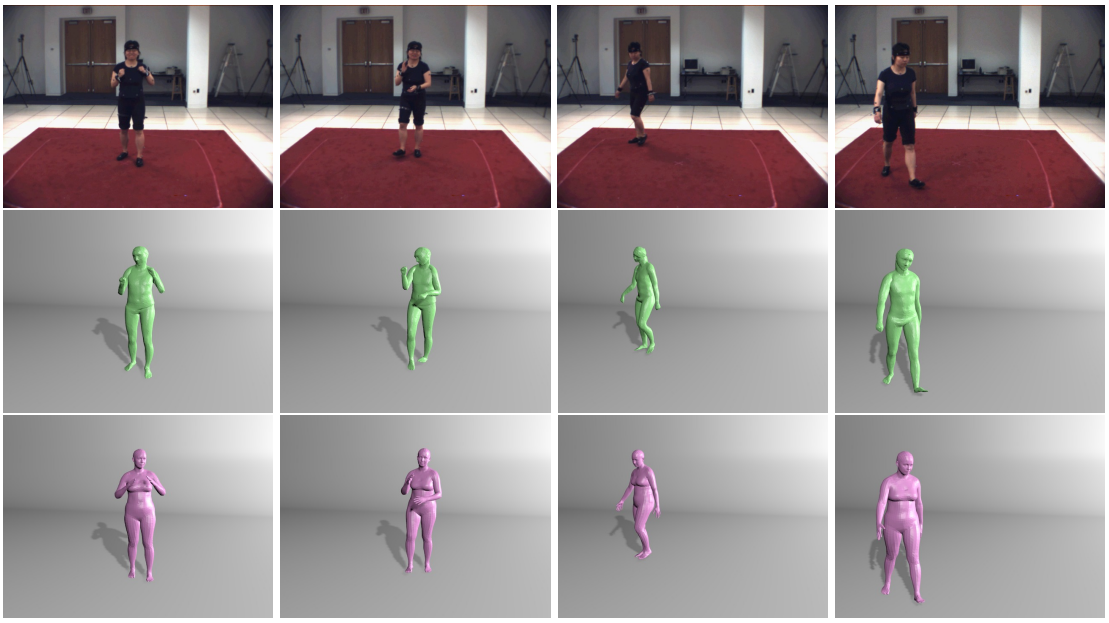


Figure 4.1: Given multi-view videos, our method can not only yield more accurate 3D pose estimation results, but also more realistic and natural meshes than the state of the art. The entire process is fully automatic. From up to bottom: example input frames; meshes returned by (Rhodin et al., 2016b); meshes generated by our method.

(a) Estimated 2D joints (b) Segmented silhouette

Figure 4.2: Automatically estimated 2D joint locations using DeepCut (Pishchulin et al., 2016) and the silhouette estimated via (Lassner et al., 2017b); here shown on the HumanEva dataset (Sigal et al., 2010).

We call the method MuVS (Multi-View SMPLify) and evaluate it quantitatively on HumanEva (Sigal et al., 2010) and Human3.6M (Ionescu et al., 2014a). We find that MuVS gives an error comparable with any published result and more realistic meshes (see Figure 4.1), while having fewer restrictions. We evaluate the method with an ablation study on HumanEva to determine which design decisions are most important.

Additionally, our approach also works in the monocular camera setting. We find that the temporal coherence term enables reasonable reconstruction of the pose from a monocular video even with a moving camera, complex background, and challenging motions. We evaluate this quantitatively on HumanEva (Sigal et al., 2010) and some challenging dancing video sequences from Youtube. The software is available for research purposes at: `https://github.com/YinghaoHuang91/MuVS`.

4.2 2D Joints and Contour Segmentation

Our method takes as input a set of 2D body joints and segmentation of the body from the background. For a direct quantitative comparison with SMPLify on standard test datasets, we use the same CNN-based joint estimator, DeepCut (Pishchulin et al., 2016). For more complex videos from the Internet, we use the real-time version of the CPM method (Cao et al., 2017) since we find it is more reliable than DeepCut. We also use a CNN trained to estimate human segmentations (Lassner et al., 2017b). Both of these are fully automatic and computed by CNNs (Pishchulin et al., 2016; Wei et al., 2016) trained on generic databases, which do not overlap with any of our test data. Illustrative joint estimation and human body segmentation results are shown in Figure 4.2.

4.3 Multi-view SMPLify

Here we first extend SMPLify to multiple camera views, then further adapt it to support temporal frames. Given the 2D joints and silhouettes for all the input frames for each camera view, we estimate the 3D pose for each time instant. We then combine information from all the views to estimate a consistent 3D human shape over time. Consequently, our algorithm is composed of two consecutive stages described in detail below.

In the first stage, a separate SMPL model is fit to all views independently at each time instant. The extension of the public SMPLify code to multiple views is straightforward: we estimate the shape and pose using information from all camera views. This gives a fully automatic approach to multi-camera marker-less motion capture. In the case of the original single-view SMPLify, the 3D pose and shape may be ambiguous given 2D joints, and the method relied heavily on priors to prevent interpenetration. In contrast, with as few as two views, many of these ambiguities go away. After that, the silhouette is used to refine the estimated shape, which is then more faithful to the observed body.

In the second stage, we first estimate a consistent 3D shape across the entire sequence by taking the median of all the shape parameters obtained in the first stage. The pose parameters for each frame are initialized with their values from the first stage. We then consider a set of consecutive frames together and regularize the motion in time. We do this by minimizing the projected joint error while encouraging the trajectory of each 3D joint to be well represented by a set of low-d DCT basis vectors (Akhter et al., 2012). This temporal smoothing helps remove errors caused by inaccurate 2D joint estimates, which may be noisy and contain errors. In particular, CNNs sometimes detect spurious points or suffer from left/right ambiguity.

4.3.1 Stage One: Per-frame Fitting

As in SMPLify, we use SMPL as our underlying shape representation. SMPL is a state-of-the-art statistical human body model (Loper et al., 2015), which is controlled by two sets of parameters, one for body shape, the other for body pose. More formally, SMPL is defined as $M(\beta,\theta;\Phi)$, where β is a vector of shape parameters that are responsible for the 3D body shape due to identity, and θ is a vector of pose parameters representing body part rotations in a kinematic tree. The fixed parameters Φ are learned from a large number of 3D body meshes. For the detailed meaning of all these parameters, we refer the reader to (Loper et al., 2015).

We first estimate the shape and pose parameters of the SMPL model for each time instant. Given the corresponding 2D joint estimates $\{J^1_{est}, J^2_{est}, \ldots, J^{|V|}_{est}\}$ for all the different views V, we formulate the energy function as the following:

$$E_M(\beta,\theta) = E_P(\beta,\theta) + \sum_{v=1}^{V} E_J(\beta,\theta;K_v,J^v_{\text{est}}), \tag{4.1}$$

where E_p is the prior term, K_v are the camera parameters for view v, and E_J is the joint fitting term (i.e. the data term). Note that here we remove the other priors used in SMPLify, because in multi-view cases, the solution is better constrained. E_P is composed of two terms: a shape prior E_β and a pose prior E_θ. The pose prior is learned from the CMU dataset (cmu, 2000), while the shape prior is learned from the SMPL body shape training data.

$$E_P(\beta,\theta) = \lambda_\theta E_\theta(\theta) + \lambda_\beta E_\beta(\beta). \tag{4.2}$$

The joint fitting term is formulated as follows:

$$E_J(\beta,\theta;K_v,J^v_{\text{est}}) = \sum_{\text{joint}\,i} w_i \rho_{\sigma_1}(\Pi_{K_v}(R_\theta(J_i(\beta))) - J^v_{\text{est},i}), \tag{4.3}$$

where $J(\cdot)$ is the joint estimation function, which returns joint locations, R is the rotation function, Π the projection function, and w_i the confidence yielded by the 2D joint detection CNN. Considering the inevitable detection noise and errors in the entire process, instead of the standard squared error, we use a robust Geman-McClure Geman (1987) error function, which is defined by:

$$\rho_\sigma(e) = \frac{e^2}{\sigma^2 + e^2}, \tag{4.4}$$

here e is the residual error, and σ is the robustness constant carefully chosen.

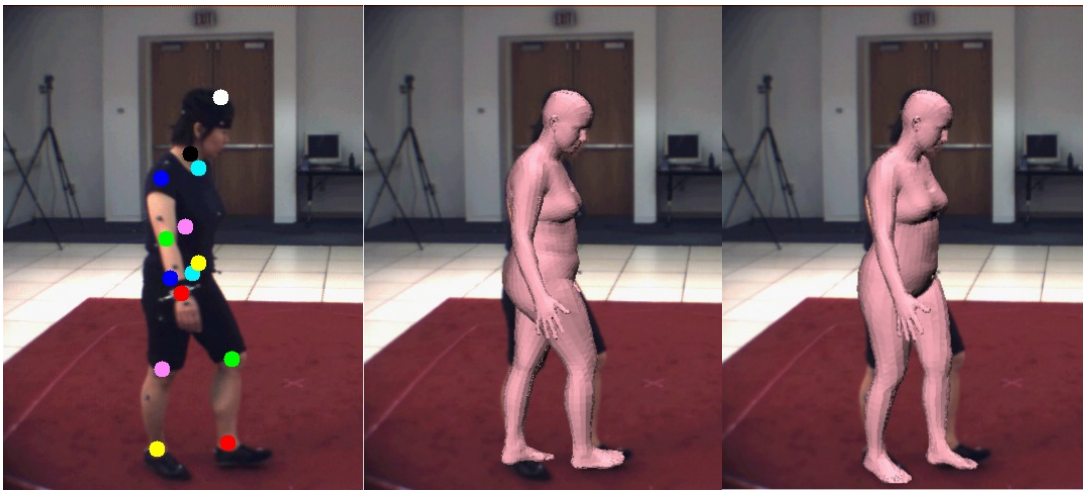
After obtaining the initial pose and shape estimation via fitting SMPL to 2D joints, we further refine it by adding silhouette information. The fitting error between the contour rendered from the SMPL model and the CNN-segmented one is defined as:

$$E_S(\beta,\theta;K_v,U_v) = \sum_{x\in\hat{S}(\beta,\theta)} l(x,U_v)^2 + \sum_{x\in U_v} l(x,\hat{S}(\theta,\beta)), \tag{4.5}$$

where $l(x,S)$ denotes the absolute distance from a point x to a silhouette S; the distance is zero when the point is inside S. The first term computes the distance from points on the projected model $\hat{S}(\beta,\theta)$ to the estimated silhouette U_v for the v-th view, while the second term computes the distance from points in the estimated silhouette U_v to the model $\hat{S}(\beta,\theta)$. As in (Lassner et al., 2017b), an L_1 distance metric is used in the second term to make it more robust to noise, while the first term uses the common L_2 distance. Combined with the 2D joint fitting term, the final energy function is:

$$E_1(\beta,\theta) = E_M(\beta,\theta) + \sum_{v\in V} E_S(\beta,\theta;K_v,U_v), \tag{4.6}$$

We found faster convergence to better solutions was obtained using a hierarchical op-



(a) (b) (c)

Figure 4.3: DCT based temporal prior helps alleviate the leg swap problem. a): Pose detection with leg swap; b): MuVS without DCT prior; c): MuVS with DCT prior.

timization strategy: firstly fitting SMPL to 2D joints can yield a coarse estimation of pose and shape parameters efficiently, then adding the silhouette fitting term can further improve accuracy.

4.3.2 Stage Two: Temporal Fitting

One obvious shortcoming of the algorithm used in the first stage is that it does not take into account the temporal relationship between consecutive frames, while in real life human motions usually present consistency. Moreover, due to the lack of texture, occlusion, similarity to the background and other noise, the joint estimator can be erroneous in ambiguous cases. One of these errors is leg swap, which is demonstrated in Figure 4.3. These errors can be difficult to correct automatically in single frame settings. By processing several consecutive frames simultaneously, we can significantly alleviate these errors.

To make our algorithm more efficient, we do not consider the silhouette in this stage and only use 2D joints. The silhouette's value resides in estimating the body shape in the first stage. We study the effect of this choice in our ablation study. Using the obtained median shape $\hat{\beta}$ and pose parameters Θ from the first stage, we optimize the following objective, which is composed of the 2D joint fitting term and low-dimensional DCT

	Walking			Boxing			Mean	Median
Method	S1	S2	S3	S1	S2	S3		
MuVS^{2}	59.22	66.81	88.60	79.51	78.68	88.34	76.86	79.10
MuVS$^{2,\,S}$	54.35	56.06	80.95	70.27	72.01	79.01	68.78	71.14
MuVS$^{2,\,S,\,T}$	50.14	56.11	79.55	68.96	71.73	78.45	67.49	70.35
MuVS$^{2,\,S,\,T,\,H}$	39.28	45.81	64.63	55.12	56.49	57.09	53.07	55.81
MuVS^{3}	52.50	62.76	82.51	72.86	73.10	80.42	70.69	72.98
MuVS$^{3,\,S}$	47.21	52.72	75.04	64.88	68.39	71.98	63.37	66.64
MuVS$^{3,\,S,\,T}$	43.11	53.37	73.56	64.00	67.94	71.44	62.23	65.97
MuVS$^{3,\,S,\,T,\,H}$	**35.51**	**44.22**	**61.30**	**49.67**	**53.89**	**51.37**	**49.33**	**50.52**

Table 4.1: Ablation results on HumanEva. 3D joint errors in *mm*. Here, labels 2/3 mean using the first 2/3 camera views; S means silhouette fitting term; T means temporal fitting term; and H means adding the silhouette fitting term at the second stage. The same notation is used in the rest of the paper.

reconstruction term B with corresponding coefficients C:

$$E_2(\Theta, C; \hat{\beta}, N) \;=\; \sum_{n=1}^{N} E_M(\hat{\beta}, \theta_n) \;+\; \sum_{\text{joint}\, e} \sum_{d \in \{X,Y,Z\}} \lambda_T E_T(C_{e,d}, D_{e,d}; \hat{\beta}, B, N), \quad (4.7)$$

here $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ is the set of pose parameters for the N frames, C are the corresponding DCT coefficients, D is the collection of all 3D SMPL joints across these frames, while $D_{e,d}$ represents the vector constructed from d-coordinate of the e-th SMPL joints, which is defined as:

$$D_{e,d} = [R_{\theta_1}(J_d(\hat{\beta}))_e, R_{\theta_2}(J_d(\hat{\beta}))_e, \ldots, R_{\theta_N}(J_d(\hat{\beta}))_e]$$

where $e \in \{1, 2, \ldots, N\}$ and $d \in \{X, Y, Z\}$. We encourage the trajectory $D_{e,d}$ across N frames to be well approximated by a low-dimensional DCT basis B:

$$E_T(c, d; \beta, B, N) = \sum_{j=1}^{N} \rho_{\sigma_2}(d_j - (Bc)_j), \quad (4.8)$$

where ρ is the same function introduced in Eq. (4.4). Note that the temporal smoothness prior is formulated on the 3D SMPL joint locations.

	Walking			Boxing			Avg
Method	S1	S2	S3	S1	S2	S3	
MuVS^{2}	18.9	19.4	20.7	19.2	13.6	20.0	18.6
MuVS$^{2,\,S}$	14.6	9.6	16.6	15.1	8.5	15.8	13.4
MuVS$^{2,\,S,\,T}$	14.1	9.3	15.9	15.0	7.4	15.1	12.8
MuVS$^{2,\,S,\,T,\,H}$	12.6	5.7	6.9	12.0	**5.4**	**7.8**	8.4
MuVS^{3}	17.6	18.6	20.6	17.2	12.3	19.4	17.6
MuVS$^{3,\,S}$	13.5	9.1	16.0	14.3	7.9	15.8	12.8
MuVS$^{3,\,S,\,T}$	13.1	8.6	15.3	14.1	5.9	14.9	12.0
MuVS$^{3,\,S,\,T,\,H}$	**12.0**	**5.5**	**6.4**	**11.3**	5.8	**7.8**	**8.1**

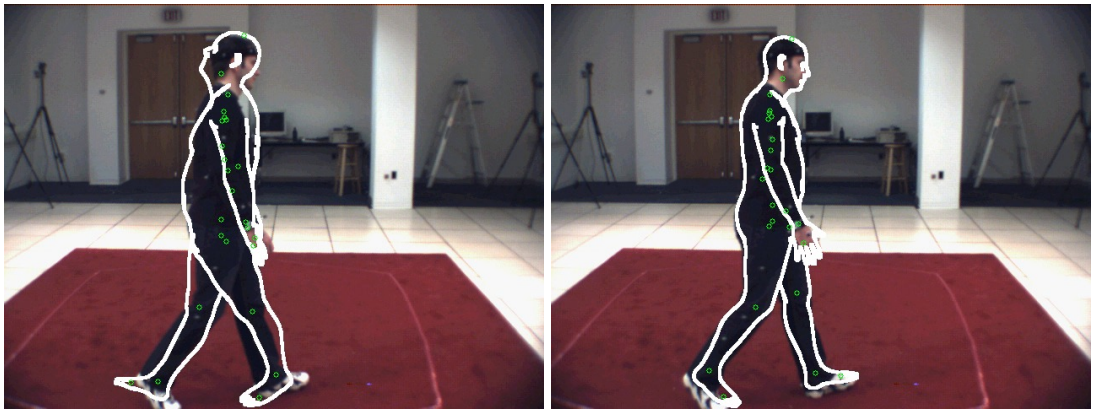
Table 4.2: Shape estimation error on HumanEva. Error in *mm*.

4.3.3 Implementation Details

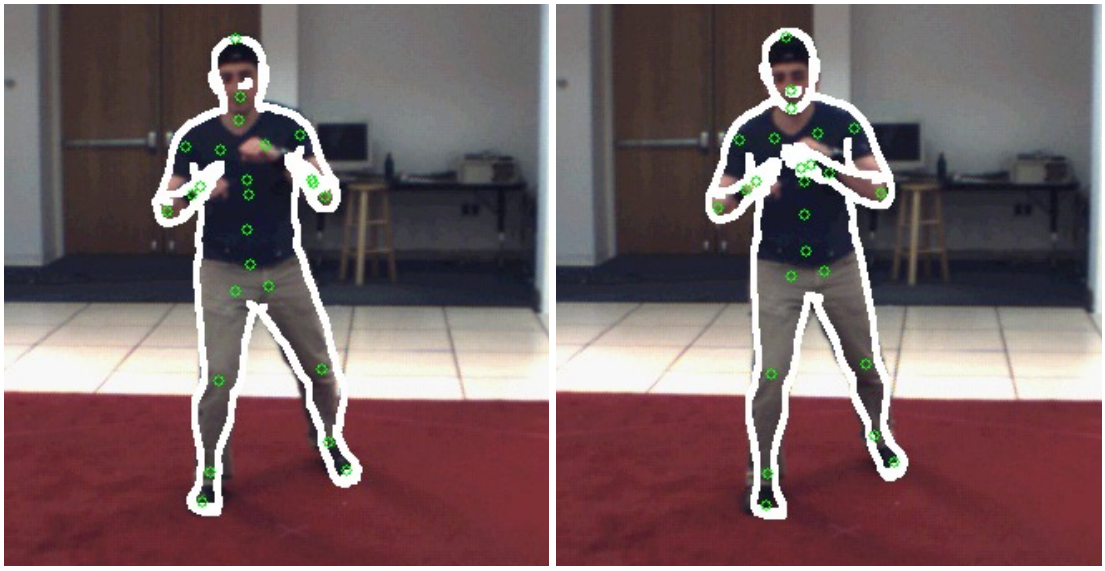
We implement our entire algorithm in Python. The two objective functions are optimized using Powell's dogleg method (Nocedal and Wright, 2006a), OpenDR (Loper and Black, 2014) and Chumpy (Chu, 2014). In the first stage, all the parameters to optimize are initialized in the same way as (Bogo et al., 2016b). For the second stage, we choose 30 consecutive frames as a unit and use the first 10 DCT components to act as the basis B. For four views with 500x500 images on a normal PC with 12GB RAM and four cores, the first stage of our method usually takes around 70 seconds for each frame, while each temporal unit in the second stage takes around 12 minutes. All the weights are empirically chosen by running our method on the training dataset of HumanEva.

4.4 Evaluation

To evaluate the effectiveness of each stage of our method, we perform experiments on two commonly used datasets, HumanEva (Sigal et al., 2010) and Human3.6M (Ionescu et al., 2014a), and compared with state-of-the-art methods (Amin et al., 2013; Belagiannis et al., 2014; Elhayek et al., 2015; Pavlakos et al., 2017b; Rhodin et al., 2016b; Sigal et al., 2012). Both datasets are collected in a controlled lab environment. HumanEva is composed of four different subjects and six different motions, while Human3.6M collects sequences from 11 subjects, each performing 15 different motions. To keep compatibility with SMPLify, we also use the first ten shape parameters in all the experiments and fine-tune all the parameters on the training dataset of HumanEva.



(a) Correct orientation error



(b) Better pose

Figure 4.4: MuVS works better than single-view SMPLify. Left column: results of SMPLify, right column: results of MuVS. The white contour represents the projected mesh.

4.4.1 Ablation Study

To analyze the effect of different parts of our algorithm, firstly, we performed various ablation experiments on HumanEva. The estimated 3D locations of joints are compared with that of the ground-truth; unlike other work, no similarity transformation is used unless stated. Error is reported in *mm* and the results are shown in Table 4.1. Note that adding the silhouette term in the second stage yields better 3D pose estimation by a large margin but at the cost of consuming much more running time. To make our algorithm comparable with other methods in running time, we do not use the silhouette term in the temporal fitting stage.

Effect of multi-view. Intuitively, multiple views can provide more information about the underlying human body. We run our algorithm on HumanEva using different numbers of views to verify this. The result indicates that adding more views consistently improves 3D pose estimation. As shown in Figure 4.4 using multiple views helps eliminate incorrectly estimated orientation and improves pose estimation accuracy.

Effect of silhouette fitting. Then, we conducted experiments to validate the effectiveness of the silhouette term in our method. Adding silhouettes consistently improves both 3D pose and shape estimation accuracy.

Effect of DCT-based temporal prior. Adding the DCT temporal smoothness term also boosts overall performance. As expected, its effect diminishes when more views are added since in this case quite good results can be obtained in the first stage.

Effect on shape estimation. To verify the effect of the aforementioned factors on body shape estimation, we run our method on the validation motion sequences of HumanEva, and compare the estimated meshes with those obtained by MoSh (Loper et al., 2014). Prior work by Loper et al. (Loper et al., 2014) shows the generated reference meshes are quite accurate. As evidenced in Table 4.2, adding silhouette information and the DCT temporal prior consistently improves body shape estimation. With three views, the average vertext-to-vertex distance is as low as 12 *mm* without the silhouette term and around 8 *mm* with it.

4.4.2 Quantitative Comparison

HumanEva: We follow the standard practice of evaluating on the "Walking" and "Boxing" sequences of subjects 1, 2, and 3. As in SMPLify (Bogo et al., 2016b), the gender of the subject is assumed known, and a gender-specific shape model is used for each motion sequence. The result is shown in Table 4.3. Here *General* means the method is trained on the training dataset of HumanEva, instead of separately training the model for each specific subject, which is referred to *Specific*. For the *General* case, we use the joint regressor distributed with SMPL to obtain 3D joints and directly compare these with the ground truth joint locations. For the *Specific* case, we use the joint regressor trained on HumanEva with MoSh, which is provided in SMPLify (Bogo et al., 2016b). Then as in (Rhodin et al., 2016b), we compute the displacement between the estimated

Figure 4.5: Monocular pose estimation results on videos downloaded from YouTube.

joint location and ground-truth in the first frame, then compensate for this difference in the remaining frames.

In the *General* case, with only two views, our method is more accurate than all the other methods using all three views. With three views, we obtain a significant improvement relative to the second-best method (55.52 vs 63.25). Our method also achieves the lowest error in the *Specific* case. Another advantage of our method over the state-of-the-art is that we return a highly realistic body mesh together with skeleton joints. Though the method proposed by Rhodin et al. (Rhodin et al., 2016b) also yields a blob-based 3D mesh, we argue that the underlying SMPL model we use is more realistic. A qualitative comparison between our results and those of (Rhodin et al., 2016b) are shown in Figure 4.1. For more results, please refer to our supplementary video at: `https://youtu.be/9iyszUxR0iw`.

Human3.6M: To further validate the generality and usefulness of MuVS, we also evaluate it on Human3.6M (Ionescu et al., 2014a). Human3.6M is the largest public dataset for pose estimation, composed of a wide range of motion types, some of them being very challenging. We use the same parameters trained on HumanEva, then apply MuVS on all the 4 views of subjects S9 and S11. We compare it with SMPLify (Bogo et al., 2016b) and other state-of-the-art multi-view pose estimation methods (Pavlakos et al., 2017b). The result is shown in Table 4.4. The multi-view version is significantly more accurate than SMPLIfy and our 3D joint estimation accuracy is quite close to that of (Pavlakos et al., 2017b), which is concurrent with our work. While they only focus on 3D joint estimation, we address 3D pose and shape estimation simultaneously. Our method returns not only 3D joint estimates, but also a realistic body shape model that is faithful to the subjects and which is ready for later modification and animation.

Method	Trained on	Walking S1	Walking S2	Walking S3	Boxing S1	Boxing S2	Boxing S3	Mean	Mean (all)
Rhodin et al. (Rhodin et al., 2016b)	General	74.9				**59.7**		67.3	
Sigal et al. (Sigal et al., 2012)		66.0						66.0	
Belagiannis et al. (Belagiannis et al., 2014)		68.3				62.7		65.5	
Elhayek et al. (Elhayek et al., 2015)		66.5				60.0		63.25	
MuVS$^{2, S, T}$		**50.14**	56.11	79.55	68.96	71.73	78.45	**60.94**	67.49
MuVS$^{3, S, T}$		**43.11**	53.37	73.56	64.00	67.94	71.44	**55.52**	62.23
Amin et al. (Amin et al., 2013)	Specific	54.5				47.7		51.10	
Rhodin et al. (Rhodin et al., 2016b)		54.6				**35.1**		44.85	
MuVS$^{3, S, T}$		**33.72**	36.78	60.11	46.85	49.92	46.99	**41.82**	45.73

Table 4.3: Quantitative comparison on HumanEva. 3D joint errors in *mm*.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sit
SMPLify (Bogo et al., 2016b)	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3
MuVS$^{4, S, T, Sim}$	**35.05**	**39.22**	**38.59**	**37.35**	**59.16**	**46.07**	**40.52**	**38.47**	**60.07**
Tekin et al. (Tekin et al., 2015)	102.41	147.72	88.83	125.28	118.02	182.73	112.38	129.17	138.89
MuVS$^{4, S, T}$	44.32	**46.99**	51.75	44.99	67.68	54.56	49.25	**48.90**	**72.82**
Pavlakos et al.(Pavlakos et al., 2017b)	**41.18**	49.19	**42.79**	**43.44**	**55.62**	**46.91**	**40.33**	63.68	97.56
	SitDown	Smoking	Waiting	WalkDog	Walk	WalkTogether	Mean	Median	
SMPLify (Bogo et al., 2016b)	137.3	83.4	77.3	79.7	86.8	81.7	82.3	69.3	
MuVS$^{4, S, T, Sim}$	**69.70**	**56.24**	**67.91**	**46.91**	**38.00**	**33.15**	**47.09**	**40.52**	
Tekin et al. (Tekin et al., 2015)	224.9	118.42	138.75	126.29	55.07	65.76	124.97	125.28	
MuVS$^{4, S, T}$	**76.51**	63.70	116.24	55.44	42.94	**37.24**	58.22	51.75	
Pavlakos et al.(Pavlakos et al., 2017b)	119.90	**52.12**	**42.68**	**51.93**	**41.79**	39.37	**56.89**	**46.91**	

Table 4.4: Quantitative comparison with SMPLify, the methods of Tekin et al. (Tekin et al., 2015) and Pavlakos et al. (Pavlakos et al., 2017b) on H3.6M dataset in *mm*. The accuracy of our method is comparable with that of the recent method proposed in (Pavlakos et al., 2017b).

	Walking			Boxing			Avg
Method	S1	S2	S3	S1	S2	S3	
SMPLify(Bogo et al., 2016b)	73.3	59.0	99.4	82.1	79.2	87.2	79.9
MuVS[1, S, Sim]	51.3	48.2	**80.9**	68.4	80.7	88.7	69.7
MuVS[1, S, T, Sim]	**51.2**	**48.1**	81.6	**61.5**	**78.3**	**82.6**	**67.2**

Table 4.5: Comparison with SMPLify on monocular videos from HumanEva in mm. Here Sim means using Procrustes analysis per frame, as with SMPLify.

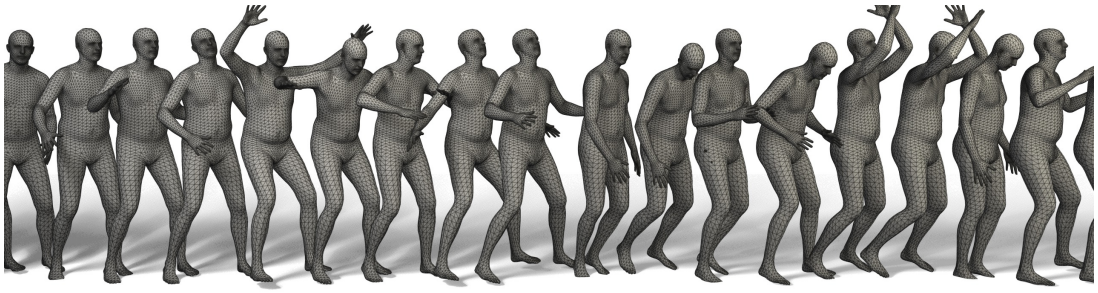


Figure 4.6: Demonstration of generated meshes for a monocular motion sequence from YouTube.

4.5 Pose and Shape from Monocular Video

Though we focus on multi-view pose and shape estimation, our method can be applied to monocular video sequences without extensive modifications while still being fully automatic. Note manually initialized pose is required for the method in (Rhodin et al., 2016b) to work on monocular data.

We compare our method with SMPLify on the first camera view of HumanEva, and the result is shown in Table 4.5. Of course, given only a single video, it is hard to apply the DCT constraint to depth, since we do not have any trustable evidence in that dimension. Empirically, our method can still return quite promising results when the performer does not move much in depth. For the videos where no camera information is provided, we manually set the focal length and other imaging parameters to some standard value as done in (Bogo et al., 2016b). We qualitatively evaluate our method on some videos downloaded from YouTube, and show the results for specific frames in Figure 4.5. Figure 4.6 shows the reconstructed mesh sequence of one of the videos. For the full video, please refer to our supplementary video at `https://youtu.be/h5ggoJlTPMI`.

4.6 Conclusion and Discussion

This chapter presents a new marker-less motion capture system, MuVS, that extends SMPLify in a principled and straightforward way. Our method computes a relatively accurate 3D pose and also returns a realistic and faithful human body mesh. Unlike previous work that assumes known silhouettes, needs user intervention, or limits the user motion, our algorithm works in a fully automatic way, and yields reasonable results even for challenging motions. Evaluation on public benchmarks validates the effectiveness and generality of our method. Additionally, we apply the approach to monocular video sequences and achieve promising results.

Future work should address more complex scenarios, like cluttered backgrounds, multiple people, and extreme poses. A pivotal direction to make the method practical is to reduce computational costs. Finally, other body parts, like faces, hands, and feet, could be easily combined into our model.

Chapter 5

Joint Human and Object Tracking

Humans constantly interact with daily objects to accomplish tasks. To understand such interactions, it is necessary to empower computers with the ability to reconstruct these from standard video cameras observing full-body interaction with scenes. Existing work struggles to capture detailed human-object interaction due to occlusion between the body and objects, motion blur, depth/scale ambiguities, and the low image resolution of hands and graspable object parts. To make the problem tractable, the community focuses either on interacting hands, ignoring the body, or on interacting bodies, ignoring hands. The GRAB dataset Taheri et al. (2020) addresses dexterous whole-body interaction but uses MoCap and lacks images, while BEHAVE Bhatnagar et al. (2022) captures video of whole-body-object interaction but lacks hand detail. We address the limitations of prior work with InterCap[1]., a novel method that reconstructs whole-body interactions with objects from multi-camera RGB-D data, using the parametric human model SMPL-X Pavlakos et al. (2019a) and known object meshes. To tackle the above challenges, InterCap uses two key observations: (i) Contact between the hand and object can be used to improve the pose estimation of both. (ii) Azure Kinect sensors allow us to set up a simple multi-view RGB-D capture system that minimizes the effect of occlusion while providing reasonable inter-camera synchronization. With this method we capture the InterCap dataset, which contains 10 subjects (5 males and 5 females) interacting with 10 objects of various sizes and affordances, including contact with the hands or feet. In total, InterCap has 223 RGB-D videos, resulting in 67,357 multi-view frames, each containing 6 RGB-D images. Our method provides quality ground-truth body meshes and objects for each video frame. Our InterCap method and dataset fill an important gap in the literature and support many research directions.

5.1 Introduction

A long-standing goal of Computer Vision is to understand human actions from images or videos. Given a single image, people effortlessly figure out what objects exist in it, the spatial layout of objects, and the pose of humans. Moreover, they deeply understand

[1]This chapter is based on the work of Huang et al. (2022)

the depicted action. What is the subject doing? Why are they doing this? What is their ultimate goal? How do they achieve this goal? To empower computers with the ability to infer such abstract concepts from pixels, we need to capture rich datasets and to devise appropriate algorithms. Since humans live in a 3D world, their physical actions involve interacting with 3D objects. Imagine how many times per day one goes to the kitchen, grabs a cup of water, and drinks from it. This involves contacting the floor with the feet, contacting the cup with the hand, moving the hand and cup together while maintaining contact, and drinking while the mouth contacts the cup. Thus, to understand human actions, it is necessary to reason in 3D about humans and objects jointly.

There is significant prior work on estimating 3D humans without taking into account the scene (Bogo et al., 2016b) and estimating 3D scenes without taking into account the human (Zollhöfer et al., 2018). There is even recent work on inserting bodies into 3D scenes such that their interactions appear realistic (Zhang et al., 2020c; Li et al., 2019; Hassan et al., 2021). But there is little work on estimating 3D humans interacting with scenes and moving objects, in which the human-scene/object contact is explicitly modeled and exploited. Moreover, to study this problem, we need a dataset of videos with rich human-object interactions with reliable 3D ground truth.

PROX (Hassan et al., 2019) took a step in this direction by estimating the 3D body in a known 3D scene. The scene mesh provides extra information that helps resolve pose ambiguities commonly encountered when a single camera is used. However, PROX involves only coarse interactions of bodies with static scenes and does not contain hand-object interaction and moving objects. The recent BEHAVE dataset (Bhatnagar et al., 2022) uses multi-view RGB-D data to capture humans interacting with objects but the dataset does not include detailed hand pose or fine hand-object contact. Finally, the GRAB dataset (Taheri et al., 2020) captures the kind of detailed hand-object and full-body-object interaction that we seek but is captured using marker-based MoCap and, hence, lacks images paired with the ground-truth scene.

We argue that what is needed is a new dataset of RGB videos containing natural human-object interaction in which the full body is tracked reliably, the hand pose is captured, objects are also tracked, and the hand-object contact is realistic; see Fig. 5.1. This is challenging and requires technical innovation to create. To that end, we design a system that uses multiple RGBD sensors that are spatially calibrated and temporally synchronized. To this data we fit the SMPL-X body model, which includes an articulated hand, by extending the PROX (Hassan et al., 2019) method to multi-view data and the SMPL-X model. We also track the 3D objects with which the person interacts. The objects used in this work are representative of items one finds in daily life and we obtain accurate 3D models for each of them with a handheld scanner. Altogether we collect 223 sequences (67, 357 multi-view frames), with 10 subjects interacting with 10 objects.

The problem, however, is that separately estimating the body and objects is not sufficient to ensure accurate 3D body-object contact. Consequently, a key innovation of this work is to estimate these jointly, while exploiting information about contact. Objects do not move independently, so, when they do, it means the body is in contact. We de-

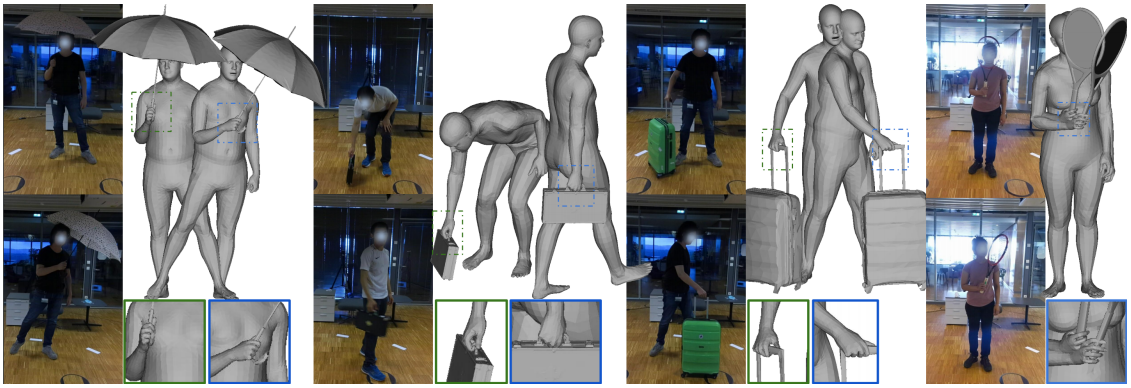


Figure 5.1: Humans interact with objects to accomplish tasks. To understand such interactions we need the tools to reconstruct them from whole-body videos in 4D, i.e., as 3Dmeshes in motion. Existing methods struggle, due to the strong occlusions, motion blur, and low-resolution of hands and object structures in such videos. Moreover, they mostly focus on the main body, ignoring the hands and objects. We develop InterCap, a novel method that reconstructs plausible interacting whole-body and object meshes from multi-view RGB-Dvideos, using contact constraints to account for strong ambiguities. With this we capture the rich InterCapdataset of 131RGB-Dvideos (38.043multi-view frames, with 6 Azure Kinects) containing 10subjects (5fe-/males) interacting with 10objects of various sizes and affordances; note the hand-object grasps.

fine likely contact regions on objects and on the body. Then, given frames with known likely contacts, we enforce contact between the body and the object when estimating the body and object poses. The resulting method produces accurate body poses, hand poses, and object poses. Uniquely it provides detailed pseudo ground-truth contact information between the body and objects in RGB video.

In summary, our major contributions are as follows: (1) We develop a novel indoor Motion Capture algorithm utilizing multiple RGB-D cameras. It is relatively lightweight, quite flexible, yet accurate enough, thus suitable for data capture of daily scenarios. (2) We extend previous work on fitting SMPL-X to images to fit it to multi-view RGB-D data while taking into account body-object contact. (3) We capture a novel dataset that contains whole-body human motions and interaction with objects, as well as multi-view RGB-D imagery.

5.2 Method

Our core goal is to accurately estimate the human motion and the object configuration throughout a video sequence. Our markerless motion capture (MoCap) method is built on top of the PROX-D approach proposed in (Hassan et al., 2019). To improve the body tracking accuracy we extend this method to use multiple RGB-D cameras (here the latest

Azure Kinect cameras). The motivation is that multiple cameras observing the body from different angles give more information about the human and object motion more than a single camera. Also commodity RGB-D cameras are much more flexible to deploy out of controlled lab scenarios than more specialized devices.

The key technical challenge lies in accurately estimating the pose and translation of the objects while a person interacts with them. In this work we focus on 10 variously sized rigid objects common in daily life like cups and chairs. Being rigid does not make the tracking of the objects trivial because of the occlusion by the body and hands. While there is a rich literature on 6 DoF object pose estimation, much of it ignores hand-object interaction. Recent work in this direction is promising but still focuses on scenarios that are significantly simpler than ours, cf. (Sun et al., 2022).

Similar to previous work on hand and object pose estimation (Hampali et al., 2020) from RGB-D videos, in this work we assume that the 3D meshes of the objects are known in advance. To this end, we first gather the 3D models of these objects from the Internet whenever possible and scan the remaining objects ourselves. To fit the known object models to image data, we first preform semantic segmentation, find the corresponding object regions in all camera views, and fit the 3D mesh to the segmented object contours via differentiable rendering. Since heavy occlusion between humans and objects in some views may make the segmentation results unreliable, aggregating segmentation from all views boosts the object tracking performance.

In the steps above, both the subject and object are treated separately and processing is per-frame, with no temporal smoothness or contact constraint applied. This produces jittery motions and heavy penetration between objects and the body. Making matters worse, our human pose estimation exploits OpenPose for 2D keypoint detection, which tends to fail when the object occludes the body or the hands interact with it. To mitigate this issue and still get reasonable hand and object pose in these challenging cases, we manually annotate the frames where the hand is in contact with the object and the hand and object vertices that are most likely to be in contact. We then explicitly encourage the labeled hand vertices to be in contact with the labeled object vertices. We find that this straightforward idea works well in practice. More details are described in the following.

5.2.1 Multi-Kinect Setup

We use multiple Azure Kinects to track the human and object together. Multiple RGB-D cameras provide a good balance between body tracking accuracy and applicability to real scenarios, compared with costly professional MoCap systems like Vicon, or cheap and convenient but not so accurate monocular RGB cameras. Moreover, this approach does not require the application of markers, making the images natural. We deploy 6 RGB-D camerasin an office room. Intrinsic camera parameters are provided by the manufacturer. To get the extrinsic camera parameters we perform camera calibration with Azure Kinect's tools (Microsoft, 2022). Given this information, we choose one camera's 3D coordinate frame as the global frame and we transform the point clouds from the other

frames into the global frame, which is where we fit the SMPL-X model.

5.2.2 Sequential Object-Only Tracking

Instance Segmentation of the Objects.

To track the objects during interaction, we need reliable cues about the object to compare with the 3D object model. To this end, we perform semantic segmentation by applying PointRend (Kirillov et al., 2020) to the whole image. We then extract the object instances that correspond to the categories of our objects. Here, we assume there is only a single object with which the subject interacts. Note that, in contrast to previous approaches where the objects occupy a large portion of the image (Hampali et al., 2020; Hassan et al., 2019; Tzionas et al., 2016; Oikonomidis et al., 2011), in our case the entire body is visible in the image, thus, the object takes up a small part of the image and is often occluded by the body and hands; our setting is much more challenging. We observe that PointRend works reasonably well for large objects like chairs, even with heavy occlusion between the object and the human, while for small objects, like a bottle or a cup, the segmentation results are significantly influenced by occlusion.

In extreme cases, it is possible for the object to not be detected in most of the views. But even when the segmentation is good, the class label for the objects may be wrong. To resolve this, we take two steps: (1) For every frame, we gather all possible object segmentation candidates and their labels. This step takes place offline and only once. (2) During the object tracking phase, for each view, we compare the rendering of the tracked object from the i^{th} frame with all the detected segmentation candidates for the $(i+1)^{\text{th}}$ frame, and preserve only the candidate with the largest overlap ratio. This rendering, comparing, and preserving operation takes place online during tracking.

Object Tracking.

Given object masks via semantic segmentation over the whole sequence, we track the object by fitting its model to observations via differentiable rendering (Kato et al., 2018; Loper and Black, 2014). This is similar to past work for hand-object tracking (Hampali et al., 2020). We assume that the object is rigid and its mesh is given. The configuration of the rigid object in the t^{th} frame is specified via a 6D rotation and translation vector ξ. Let R_S and R_D be functions that render a synthetic mask and depth image for the tracked 3D object mesh, M. Let also $S = \{S_v\}$ be the "observed" object masks and $D = \{D_v\}$ be corresponding depth values for the current frame, where v is the camera view. Then, we minimize:

$$\begin{aligned} E_O(\xi; S, D) = \sum_{\text{view } v} & \lambda_{segm} \| (R_S(\xi, M, v) - S_v) * S_v \|_F^2 + \\ & \lambda_{depth} \| (R_D(\xi, M, v) - D_v) * S_v \|_F^2, \end{aligned} \tag{5.1}$$

where the two terms compute how well the rendered object mask and depth match the detected ones; the $*$ is an element-wise multiplication, and $\|.\|_F$ the Frobenius norm. For simplicity, we assume that transformations from the master to other camera frames are encoded in the rendering functions R_S, R_D; we do not denote these explicitly here.

5.2.3 Sequential Human-Only Tracking

We estimate body shape and pose over the whole sequence from multi-view RGB-D videos in a frame-wise manner. This is similar in spirit with the PROX-D method (Hassan et al., 2019), but, in our case, there is no 3D scene constraint and multiple cameras are used. The human pose and shape are optimized independently in each frame. We use the SMPL-X (Pavlakos et al., 2019a) model to represent the 3D human body. SMPL-X is a function that returns a water-tight mesh given parameters for shape, β, pose, θ, facial expression, ψ, and translation, γ. We follow the common practice of keeping the dimension of β as 10 and use the 32-dimension latent space in VPoser (Pavlakos et al., 2019a) to represent body pose.

We minimize the loss defined below. For the current t^{th} frame, we essentially extend the major loss terms used in PROX (Hassan et al., 2019) to multiple views:

$$\begin{aligned} E_B(\beta,\theta,\psi,\gamma;K,J_{est}) = E_J + \lambda_D E_D + \lambda_{\theta_b} E_{\theta_b} + \lambda_{\theta_f} E_{\theta_f} + \\ \lambda_{\theta_h} E_{\theta_h} + \lambda_\alpha E_\alpha + \lambda_\beta E_\beta + \lambda_\mathcal{E} E_\mathcal{E} + \lambda_\mathcal{P} E_\mathcal{P}, \end{aligned} \tag{5.2}$$

where E_β, E_{θ_b}, E_{θ_h}, E_{θ_f}, $E_\mathcal{E}$ are prior loss terms for body shape, body pose, hand pose. facial pose and expressions. Also, E_α is a prior that penalizes extreme elbow and knee bending. For detailed definitions of these terms see (Hassan et al., 2019). E_J is a 2D keypoint re-projection loss:

$$E_J(\beta,\theta,\gamma;K,J_{\text{est}}) = \sum_{view\ v} \sum_{joint\ i} k_i^v w_i^v \rho_J\big(\Pi_K^v\big(R_{\theta\gamma}(J(\beta)_i)\big) - J_{est,i}^v\big), \tag{5.3}$$

where v and i iterate through views and joints, k_i^v and w_i^v are the per-joint weight and detection confidence, ρ is a robust Geman-McClure error function (Geman and McClure, 1987), Π_K^v is the projection function with K camera parameters, $R_{\theta\gamma}(J(\beta)_i)$ are the posed 3D joints of SMPL-X, and $J_{est,i}^v$ the detected 2D joints. The term E_D is:

$$E_D(\beta,\theta,\gamma,K) = \sum_{view\ v} \sum_{p\in P^v} \min_{v\in V_b^v} \|v-p\|, \tag{5.4}$$

where P^v is Azure Kinect's segmented point cloud, and V_b^v are SMPL-X vertices that are visible in the v^{th} view. This term measures how far the estimated body mesh is from the combined point clouds, i.e., it is a multi-view extension of PROX's (Hassan et al., 2019) loss. Note that, unlike PROX, we have multiple point clouds from all views. For each view we dynamically compute the visible body vertices, and "compare" them against the point cloud gathered for that view.

Figure 5.2: The objects of our InterCap dataset. **Top:** Color photos. **Bottom:** Annotations for object areas that are likely to come in contact during interaction, shown in red.

Finally, the term $E_{\mathcal{P}}$ penalizes self-interpenetration of the SMPL-X body mesh; see PROX (Hassan et al., 2019) for a more detailed and formal definition of this:

$$E_{\mathcal{P}}(\theta,\beta,\gamma) = E_{\mathcal{P}_{self}}(\theta,\beta). \tag{5.5}$$

5.2.4 Joint Human-Object Tracking Over All Frames

We treat the result of the above optimization as initialization for refinement via joint optimization of the body and the object over all frames, subject to contact constraints.

For this we fix the body shape parameters as the mean body shape computed over all frames from the first stage, as done in (Huang et al., 2017). Then, we jointly optimize the object and body pose and translation over all frames. We add a temporal smoothness loss to reduce jitter for both the human and the object. We also penalize the body-object interpenetration, as done in PROX (Hassan et al., 2019). A key difference is that, in PROX, the scene is static, while, in our case, the object is free to move in space. To enforce contact, we annotate the body, feet and hand parts that are most likely to be in contact with the objects and, for each object, we label vertices most likely to be contacted. These annotations are shown in Fig. 5.3 and Fig. 5.2-bottom, respectively, in red. We also annotate frame sub-sequences where hands are in contact with objects, and enforce

contact between them explicitly to get reasonable tracking results even when there is heavy interaction and occlusion between hands and objects. Such interactions prove to be challenging for current state-of-the-art 2D hand joint detectors, e.g., OpenPose.

Formally, we perform global optimization over all T frames, and minimize a loss E that is composed of an object fitting loss, E_O, a body fitting loss, E_B, a motion smoothness prior (Zhang et al., 2021a) loss, $E_{\mathcal{A}}$, and a loss penalizing object acceleration, $E_{\mathcal{B}}$. We also use a ground support loss, $E_{\mathcal{G}}$, that encourages the human and the object to be above the ground plane, i.e., to not penetrate it. Last, we use a hand-object contact loss, $E_{\mathcal{C}}$, that attaches the hand to the object for frames with contact. The loss E is defined as:

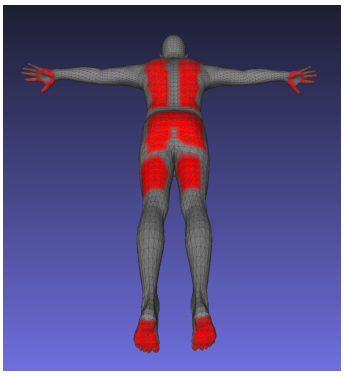


Figure 5.3: Annotation of likely body contact areas (red color).

$$
\begin{aligned}
E = \; & \lambda_{\mathcal{A}} E_{\mathcal{A}}(\Theta,\Psi,\Gamma,A;\beta^*,T) + \lambda_{\mathcal{B}} E_{\mathcal{B}}(\Xi,T,M) + \\
& \frac{1}{T}\sum_{\text{frame } t}\Big[E_O(\Xi_t;\mathcal{S}_t,\mathcal{D}_t) + E_B(\beta^*,\Theta_t,\Psi_t,\Gamma_t;\mathcal{J}_{est})\Big] + \\
& \frac{1}{T}\sum_{\text{frame } t}\Big[E_{\mathcal{P}}(\Theta_t,\beta^*,\Gamma_t) + E_{\mathcal{C}}(\beta^*,\Theta_t,\Psi_t,\Gamma_t,\Xi_t,M)\Big] + \\
& \frac{\lambda_{\mathcal{G}}}{T}\sum_{\text{frame } t}\Big[E_{\mathcal{G}}(\beta^*,\Theta_t,\Psi_t,\Gamma_t) + E_{\mathcal{G}'}(\Xi_t,M)\Big] + \\
& \frac{\lambda_{\mathcal{Q}}}{T}\sum_{\text{frame } t}\Big[Q_t * E_{\mathcal{C}}(\beta^*,\Theta_t,\Psi_t,M',\Xi_t)\Big],
\end{aligned}
\tag{5.6}
$$

where E_O comes from Eq. (5.1) and E_B from Eq. (5.2), and both go through all views v, while $E_{\mathcal{P}}$ comes from Eq. (5.5). For all frames $t = \{1,\ldots,N\}$ of a sequence, $\Theta = \{\theta_t\}$, $\Psi = \{\psi_t\}$, $\Gamma = \{\gamma_t\}$, are the body poses, facial expressions and translations, $\Xi = \{\xi_t\}$ is the object rotations and translations, $\mathcal{S} = \{S_t\}$ and $\mathcal{D} = \{D_t\}$ are masks and depth patches, and $\mathcal{J}_{est} = \{J_{est,t}\}$ are detected 2D keypoints. M is the object mesh, and β^* the mean body shape. $E_{\mathcal{C}}$ encourages hand and object contact for frames in contact, which are indicated by the manually annotated binary vectors $Q = \{Q_1, Q_2, \ldots, Q_T\}$. Here Q_t is set to 1 if in the t^{th} frame one of the hands is in contact with the object, and set to 0 otherwise. The formal definition of $E_{\mathcal{C}}$ is:

$$
\begin{aligned}
E_{\mathcal{C}}(\beta^*,\Theta_t,\Psi_t,\Gamma_t,\Xi_t,M) = CD\Big(& H\big(W(\Theta_t,\Psi_t,\Gamma_t,A,\beta^*)\big), \\
& H'\big(W'(\Xi_t,M)\big)\Big),
\end{aligned}
\tag{5.7}
$$

where CD refers to the Chamfer Distance function, H is a function that returns only the annotated body-contact vertices of Fig. 3, H' returns the closest points on the object for these body-contact vertices, W' deforms rigidly the object and is explained in the previous paragraph and W similarly (non-rigidly) deforms the SMPL-X mesh and concatenates the vertices into a single vector. The correspondences between the body (hand) vertices and object vertices are established via the Iterative-Closest-Point (ICP) method

Name	#Motions	Natural Appearance	Moving Objects	Accurate Motion	With Image	Articulated Hands
HumanEva	56	✓	✗	✓	✓	✗
Human3.6M	165	✓	✗	✓	✓	✗
AMASS	11265	✓	✗	✓	✗	✗
GRAB	1334	✓	✓	✓	✗	✓
3DPW	60	✓	✗	✓	✓	✗
GTA-IM	119	✗	✗	✓	✓	✗
SAIL-VOS	201	✗	✗	✗	✗	✗
PiGraphs	63	✓	✗	✓	✓	✗
PROX	20	✓	✗	✗	✓	✗
RICH	142	✓	✗	✓	✓	✗
BEHAVE	321	✓	✓	✓	✓	✗
InterCap	223	✓	✓	✓	✓	✓

Table 5.1: Dataset statistics. Comparison of our InterCapdataset to existing datasets.

Besl and McKay (1992) and dynamically updated during the optimization. The motion smoothness loss $E_{\mathcal{A}}$ penalizes abrupt position changes for body vertices, and the vertex acceleration loss $E_{\mathcal{B}}$ encourages smooth object trajectories. We estimate the ground plane surface by fitting a plane to chosen floor vertices. The terms $E_{\mathcal{G}}$ and $E_{\mathcal{G}'}$ measure whether the body and object penetrate the ground, respectively. We use PyTorch for the implementation, and adopt L-BFGS Nocedal and Wright (2006b) with strong Wolfe line search as the solver.

5.3 InterCap Dataset

We use the proposed InterCap algorithm (Sec. 5.2) to capture the InterCap dataset, which uniquely features whole-body interactions with objects in multi-view RGB-D videos.

Data-capture protocol: We use 10 objects shown in Fig. 5.2-top that vary in size and "afford" different interactions with the body, hands or feet. We recruit 10 subjects (5 males and 5 females) that are 25-40 years old. The subjects are recorded interacting with 7 or more objects. Subjects are instructed to interact with objects as naturally as possible. However, as Azure Kinects supports only up to 30 FPS, we avoid very fast interactions that cause severe motion blur. We capture two or three sequences per object. We also capture each subject performing a freestyle interaction of their choice. All subjects gave informed written consent to share their data for research.

4D reconstruction: Our InterCap method (Sec. 5.2) takes as input multi-view 3D videos and outputs 4D meshes for the human and object, i.e., 3D meshes over time. Humans are represented as SMPL-X meshes (Pavlakos et al., 2019a), while object meshes are acquired with an Artec hand-held scanner (Artec 3D, Luxembourg). Some dataset frames along with the reconstructed meshes are shown in Fig. 5.1 and Fig. 5.4; see also our video at: `https://youtu.be/d5wHLDIqN6c`. Reconstructions look natural, with plausible contact between the human and the object.

Dataset statistics: InterCap has 223 RGB-D videos with a total of multi-view frames (6 RGB-D images each). For a comparison with other datasets, see Tab. 5.1.

5.4 Experiments

Contact heatmaps: Figure 5.5-top shows contact heatmaps on each object, across all subjects. We follow the protocol of GRAB (Taheri et al., 2020), which uses a proximity metric on reconstructed human and object meshes. First, we compute per-frame binary contact maps by thresholding (at 4.5mm) the distances from each body vertex to the closest object surface point. Then, we integrate these maps over time (and subjects) to get "heatmaps" encoding contact likelihood. InterCap reconstructs human and object meshes accurately enough so that contact heatmaps agree with object affordances, e.g., the handle of the suitcase, umbrella and tennis racquet are likely to be grasped, the upper skateboard surface is likely to be contacted by the foot, and the upper stool surface by the buttocks. Figure 5.5-bottom shows heatmaps on the body, computed across all subjects and objects. Heatmaps show that most interactions involve mainly the right hand. Contact on the palm looks realistic, concentrated on the fingers and MCP joints. Contact on the dorsal side is due to our challenging setup and some reconstruction jitter.

Penetration metrics: We evaluate the penetration between human and object meshes for all sequences of our InterCap dataset. We follow the protocol of GRAB et al. (Taheri et al., 2020); we first find the "contact frames" for which there is at least minimal human-object contact, and then report statistics for these. In Fig. 5.6-left we show the distribution of penetrations, i.e., the number of "contact frames" with a given mesh penetration depth. In Fig. 5.6-right we show the cumulative distribution of penetration, i.e., the percentage of "contact frames" for which mesh penetration is below a threshold. Roughly 60% of "contact frames" have $\leq 5mm$, 80% $\leq$ 7 mm, and 98% $\leq$ 20 mm penetration. The average penetration depth over all "contact frames" is 7.2 mm.

Fitting accuracy: For every frame, we compute the distance from each mesh vertex to the closest point-cloud (PCL) point; for each human or object mesh we take into account only the respective PCL area obtained with PointRend (Kirillov et al., 2020) segmentation. The mean vertex-to-PCL distance is 20.29mm for the body, and 18.50mm for objects. In comparison, PROX-D (Hassan et al., 2019), our base method, achieves an error of 13.02mm for the body. This is expected since PROX-D is free to change the body shape to fit each individual frame, while in our method estimates a single body shape for the whole sequence. SMPLify-X (Pavlakos et al., 2019a) achieves an mean error of 79.54mm, for VIBE the mean error is 55.59mm, while ExPose gets an mean error of 71.78mm. These numbers validate the effectiveness of our method for body tracking. Note that these methods are based on monocular RGB images only, so there is not enough information for them to accurately estimate the global position of the 3D body meshes. Thus we first align the output meshes with the point clouds, then compute the error. Note that the error is bounded from below for two reasons: (1) it is influenced by factory-design imperfections in the synchronization of Azure Kinects, and (2) some vertices reflect body/object areas that are occluded during interaction and their closest PCL point is a wrong correspondence. Despite this, InterCap empirically estimates reasonable bodies, hands and objects in interaction, as reflected in the contact heatmaps and

penetration metrics discussed above.

Ablation of contact term: Figure 5.7-top shows results with-/out our term that encourages body-object contact; images show detail of hand-object grasps. The results show that encouraging contact yields more natural hand poses and fewer inter-penetrations. This is backed up by the contact heatmaps and penetration metrics discussed above.

Ablation of temporal smoothing term: Figure 5.7-bottom shows results with-/out our temporal smoothing term. Each solid line shows the acceleration of a randomly chosen vertex, without the temporal smoothness term; we show 3 different motions. The dashed lines of the same color show the same motions with the smoothness term; these are clearly smoother. We find that the learned motion prior of Zhang et al. (Zhang et al., 2021a) produces a more natural motion than handcrafted ones (Huang et al., 2017). However, results still have some jitter, especially for hands, due to their low image resolution, motion blur, and coarse point clouds. Future work should study more advanced motion priors.

5.5 Discussion

In this work, we focus on full-body human interaction with everyday rigid objects. We present a novel method, called InterCap, that reconstructs such interactions from multi-view full-body videos, including natural hand poses and contact with objects. With this method, we capture the InterCap dataset, containing a variety of human subjects interacting with several common objects. The dataset contains reconstructed 3D meshes for the subject and object over time, as well as plausible contacts between them. In contrast to most previous work, our method uses no special devices like optical markers or IMUs, but only several consumer-level RGB-D cameras. Our setup is lightweight and has the potential of being adopted in daily scenarios. Our method estimates reasonable hand poses even when there is heavy occlusion between hands and the object. One limitation of our method is the need to manually annotate the frames where the subject and the object are in interaction. This procedure can be tedious and time-consuming when the size of the dataset is large, thus it will be interesting to explore various heuristics to automatically detect contact frames. In future work, we also plan to study interactions with smaller objects and dexterous manipulation. We also plan to study interactions with smaller objects and dexterous manipulation. Our data and code are available at https://intercap.is.tue.mpg.de.

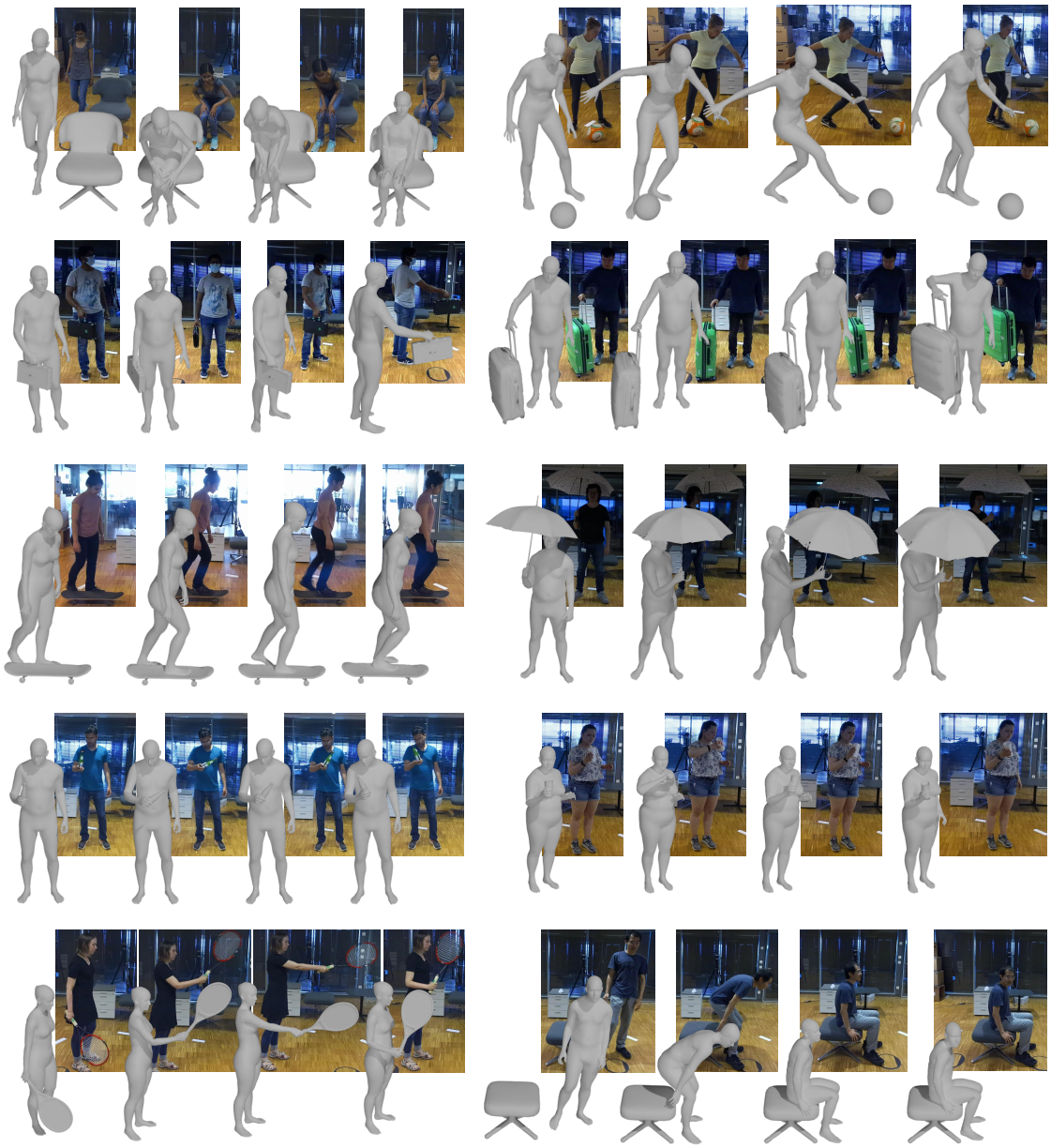


Figure 5.4: Samples from our InterCap dataset, drawn from four sequences with different subjects and objects. The estimated 3D object and SMPL-X human meshes have plausible contacts and agree with the input RGB images. Best viewed zoomed in.

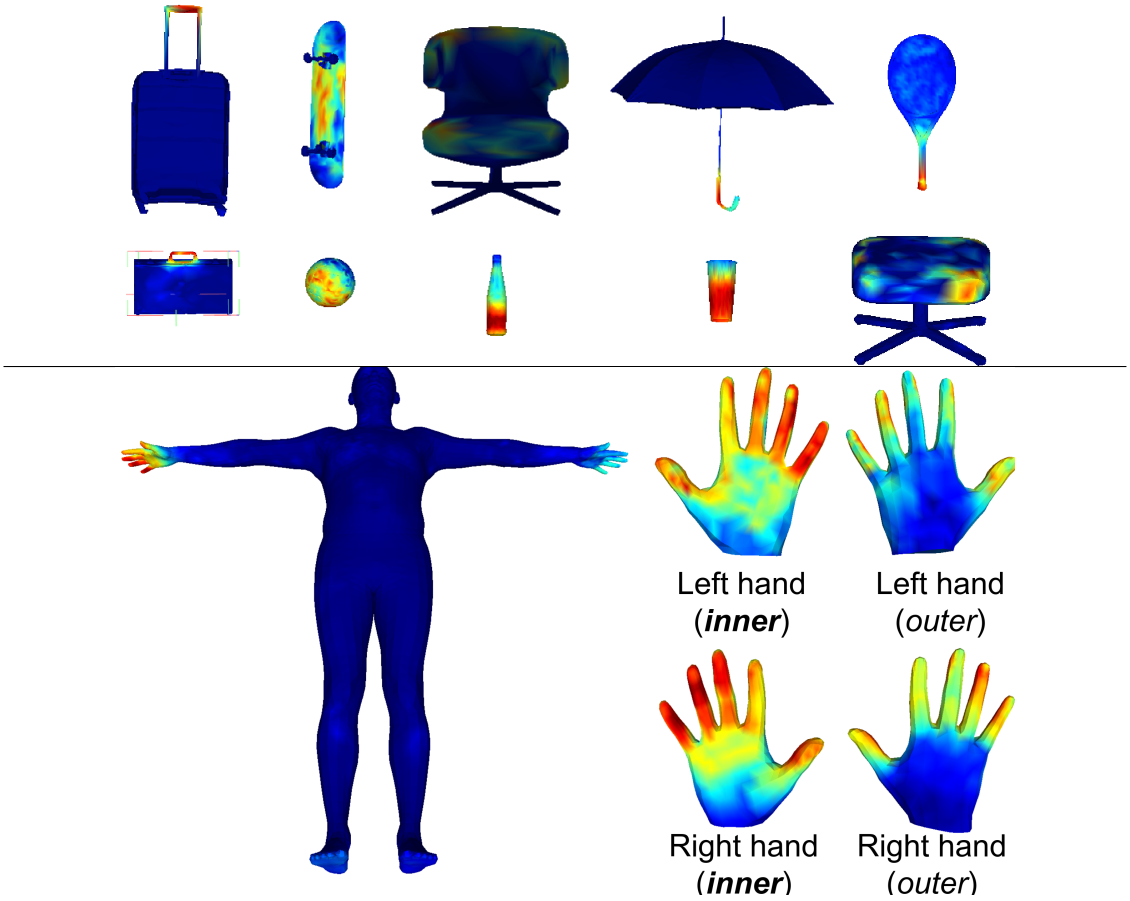


Figure 5.5: Contact heatmaps for objects (across all subjects) and the human body (across all objects/subjects). High contact likelihood is shown with red color, and low with blue. Color-coding is normalized separately for each object, the body, and each hand.

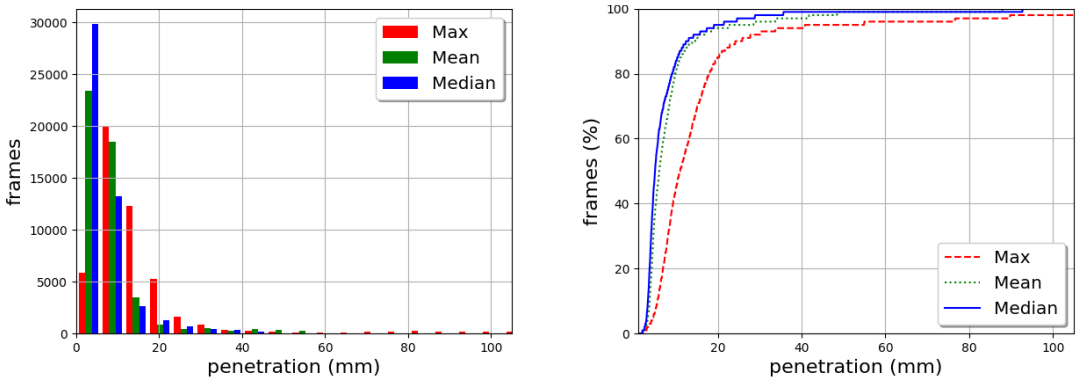


Figure 5.6: Statistics of human-object mesh penetration for all InterCap sequences. **(Left)** The number of frames (Y-axis) with a certain penetration depth (X-axis). **(Right)** The percentage of frames (Y-axis) with a penetration depth below a threshold (X-axis).

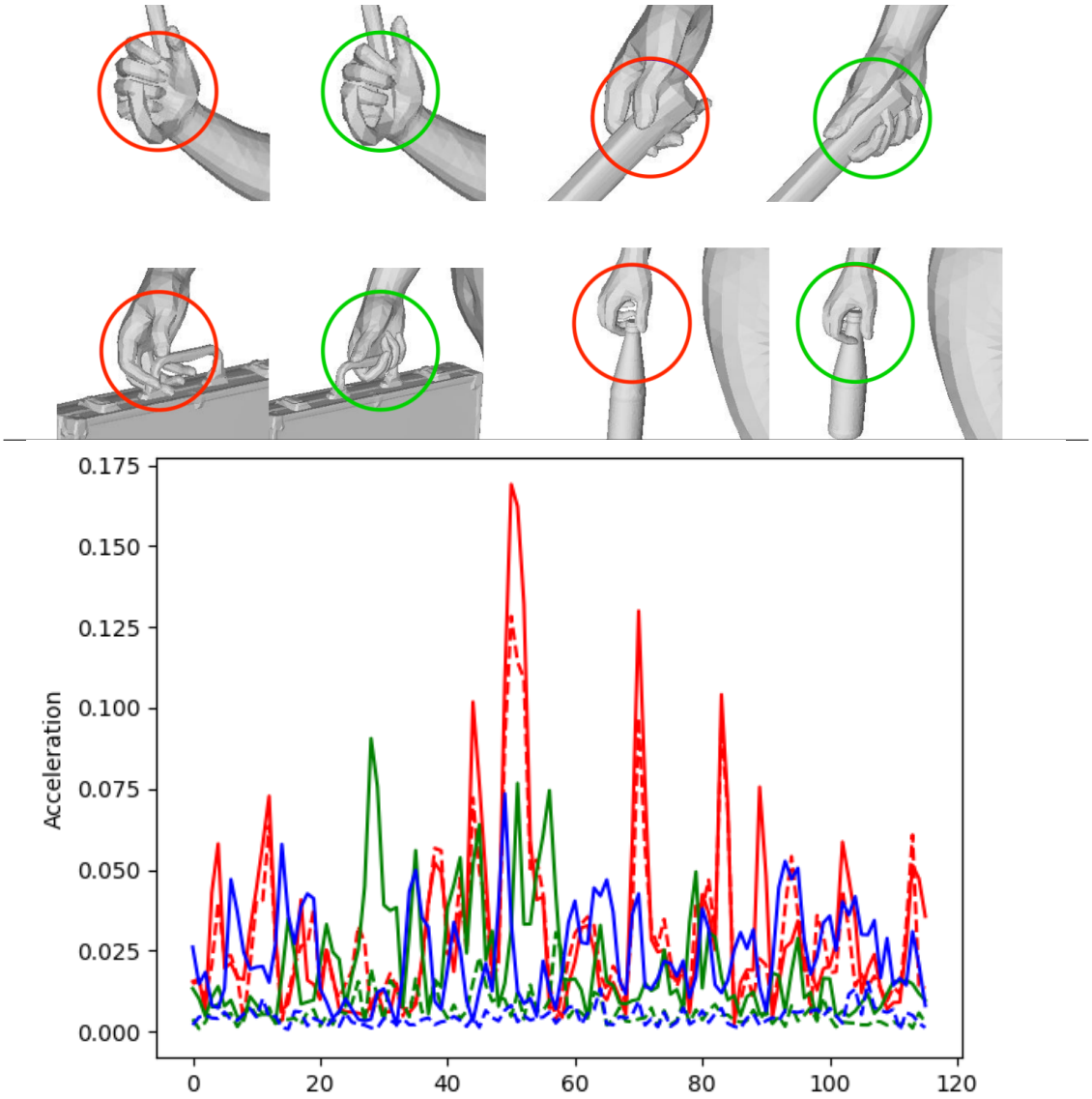


Figure 5.7: **(Top)** Qualitative ablation of our contact term. Each pair of images shows results wo/ (red) and w/ (green) the contact term. Encouraging contact results in more natural hand poses and hand-object grasps. **(Bottom)** Acceleration of a random vertex w/ (dashed line) and wo/ (solid line) temporal smoothing for 3 motions over the first 120 frames.

Chapter 6

Real-time MoCap from Sparse IMUs

6.1 Introduction

Many applications such as gaming, bio-mechanical analysis, and emerging human-computer interaction paradigms such as Virtual and Augmented Reality (VR/AR) require a means to capture a user's 3D skeletal configuration. Such applications impose three challenging constraints on pose reconstruction: (i) it must operate in real-time, (ii) it should work in everyday settings such as sitting at a desk, at home, or outdoors, and (iii) it should be minimally invasive in terms of user instrumentation. However no single public available system satisfies all three aforementioned requirements. Please refer to Chapter 3 for a detailed discussion of the pros and cons of the current state-of-the-art solutions.

Given that emerging consumer products such as smart-watches, fitness trackers and smart-glasses (e.g., HoloLens, Google Glass) already integrate IMUs, reconstructing 3D body pose from a small set of sensors in real-time would enable many applications. In this chapter we introduce DIP: Deep Inertial Poser[1], the first deep learning method capable of estimating 3D human body pose from only 6 IMUs in real time. Learning a function that predicts accurate poses from a sparse set of orientation and acceleration measurements alone is a challenging task because (i) the whole pose is not observable from just 6 measurements, (ii) previous work has shown that long-range temporal information plays an important role Von Marcard et al. (2017), and (iii) capturing large datasets for training is time-consuming and expensive.

To overcome these issues we leverage the following observations and insights: (i) Large datasets of human Mocap such as CMU De la Torre et al. (2008) or the Human3.6M Ionescu et al. (2014b) exist. Specifically, we leverage AMASS Mahmood et al. (2019), a large collection of MoCap datasets with data provided in the form of SMPL Loper et al. (2015) model parameters. We leveraged this to synthesize IMU data. Specifically, we place virtual sensors on the SMPL mesh and use the Mocap sequences to obtain virtual orientations via forward kinematics, and accelerations via finite differences. We leverage this synthetic data for training a deep neural network model. (ii) To model long-range temporal dependencies, we leverage recurrent neural networks (RNNs) to map from orientations and accelerations to SMPL parameters. However, making full

[1]This chapter is based on the work of Huang et al. (2018).

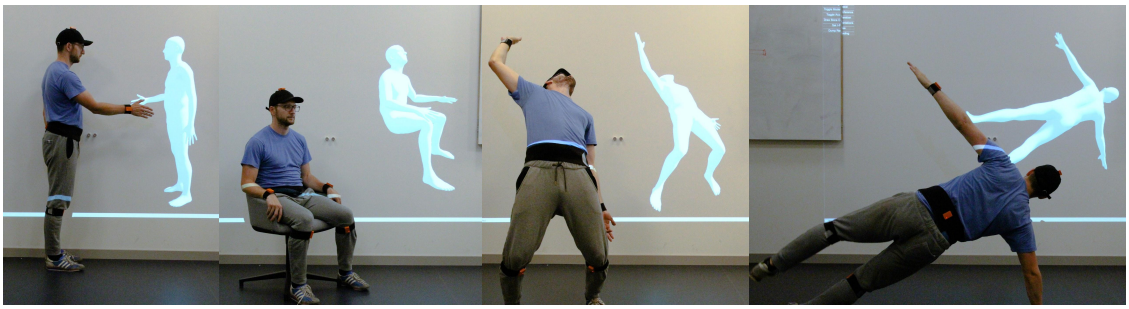


Figure 6.1: We demonstrate a novel learning-based method for reconstructing 3D human pose in real-time using only six body-worn IMUs. Our method learns from a large synthetic dataset and at runtime predicts pose parameters from real IMU inputs in real-time while only requiring minimal user instrumentation. This brings 3D motion capture to new scenarios that are difficult for camera-based methods such as heavy occlusions and fast motion.

use of acceleration information proved to be difficult. This leads to systematic errors for ambiguous mappings between sensor orientation (measured at the lower-extremities) and pose. In particular knee and arm bending is problematic. To alleviate this issue we introduce a novel loss-term that forces the network to reconstruct accelerations during training, which preserves information throughout the network stack and leads to better performance at test time. (iii) Offline approaches for the same task leverage both past and future information Von Marcard et al. (2017). We propose an extension of our architecture that leverages bi-directional RNNs to further improve the reconstruction quality. At training time, this architecture has access to the same information as Von Marcard et al. (2017), and propagates information from the past to the future and vice versa. To retain the real-time regime, we deploy this architecture at test time in a sliding-window fashion. We experimentally show that only 5 future frames are sufficient for high-quality predictions, while only incurring a modest latency penalty of 85ms.

Using only synthetic data for training already provides decent performance. However, real IMU data contains noise and drift. To close the gap between synthetic and real data distributions, we fine-tune our model using a newly created DIP-IMU dataset containing approximately 90 minutes of real IMU data.

We experimentally evaluate DIP using TotalCapture Trumble et al. (2017), a benchmark dataset including IMU data and reference ("ground truth") poses, and on the DIP-IMU dataset. We show that DIP achieves an accuracy of 15.85° angular error, which is lower than the competing offline approach, SIP Von Marcard et al. (2017). This is significant as our method runs in real-time, whereas SIP requires the full motion sequence as input. To further demonstrate the real-time capabilities of DIP, we integrate our approach in a simple VR proof-of-concept demonstrator; we take raw IMU data as input and our pipeline predicts full body poses, without any temporal filtering or post-hoc processing.

The resulting poses are then visualized via Unity. In summary, DIP occupies a unique place in the pose estimation literature as it satisfies all three aforementioned constraints: it is real time, minimally intrusive, and works in everyday places Fig. 6.1.

6.2 Method Overview

Our goal is to reconstruct articulated human motion in unconstrained settings from a sparse set of IMUs (6 sensors) in real-time. This problem is extremely difficult since many parts of the body are not directly observable from the sensor data alone. To overcome this problem, we leverage a state-of-the-art statistical model of human shape and pose, and regress its parameters using a deep recurrent neural network (RNN). In our implementation, we use the SMPL model Loper et al. (2015) both to synthesize training data and as the output target of the LSTM architecture. This approach ensures that sufficient data is available for training, and encourages the resulting predictions to natural human motion. We now briefly introduce the most salient aspects of the data generation process (Sec. 6.2.1, Sec. 6.2.2), the accumulated dataset used for training (Sec. 6.2.3), and our proposed network architecture (Sec. 6.2.4). An overview of the entire pipeline can be found in Fig. 6.2.

6.2.1 Background: SMPL Body Model

We choose SMPL body model as the output representation for its high realism, great expressiveness and fast computation speed. For a overall discussion about the SMPL body model and its variants, please refer to Section 2.1 in Chapter 2.

6.2.2 Synthesizing Training Data

Our approach is learning-based and hence requires a sufficiently large dataset for training. Compared to the camera or marker-based cases, there are very few publicly available datasets including IMU data and ground-truth poses. To the best of our knowledge the only such dataset is TotalCapture Trumble et al. (2017), including typical day-to-day activities. The dataset contains synchronized IMU, Mocap and RGB data. However, due to the limited size and types of activities, models trained on this dataset alone do not generalize well, e.g., common interaction tasks in VR/AR such as reaching for and selecting objects in a seated position are not represented at all.

However, given the capability of fitting SMPL parameters to inputs of different modalities (17 IMUs, marker data), it becomes feasible to generate a much larger and more comprehensive training dataset by synthesizing pairs of IMU measurements and corresponding SMPL parameters from a variety of input datasets.

To attain synthetic IMU training data, we place virtual sensors on the SMPL mesh surface. The orientation readings are then directly retrieved using forward kinematics,

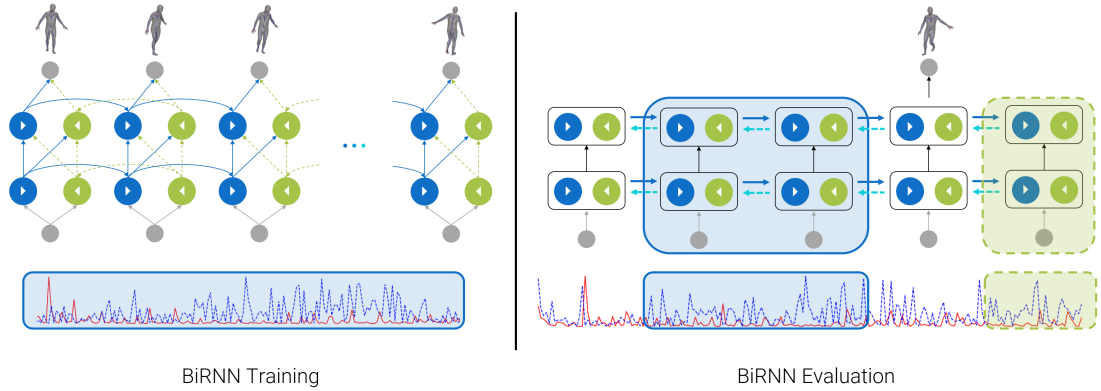


Figure 6.2: Overview: Left: At training-time our network has access to the whole sequence (blue window) and propagates temporal information from the past to the future and vice versa. Our model consists of two stacked bidirectional layers. Shown in blue (solid arrows) is the forward layer and in green (dashed arrows) the backward layer. Note that the second layer receives direct input from the forward and backward cells of the layer below (diagonal arrows). Please refer to Fig. 6.4, for more details. Right: At runtime we feed a sliding window of short subsequences from the past (blue window) and the future (green window) to predict the pose at the current time step. This incurs only minimal latency and makes real-time applications feasible.

whereas we obtain accelerations via finite differences. Assuming the position of a virtual IMU is p_t for time t, and the time interval between two consecutive frames is dt, the simulated acceleration is computed as:

$$a_t = \frac{p_{t-1} + p_{t+1} - 2 * p_t}{dt^2}. \tag{6.1}$$

6.2.3 Datasets

Our final training data is a collection of pairs of synthetic IMU sensor readings and corresponding SMPL pose parameters. We use a subset of the AMASS dataset Mahmood et al. (2019), itself a combination of datasets from the computer graphics and vision literature, including CMU De la Torre et al. (2008), HumanEva Sigal et al. (2010), JointLimit Akhter and Black (2015), and several smaller datasets.

We use two further datasets (TotalCapture and DIP-IMU) for evaluation of our method. Both consist of pairs of *real* IMU readings and reference ("ground-truth") SMPL poses. To obtain reference SMPL poses for TotalCapture Trumble et al. (2017), we used the method of Loper et al. (2014) on the available marker information. Finally, we recorded the DIP-IMU dataset using commercially available XSens sensors. The corresponding SMPL poses were obtained by running SIP Von Marcard et al. (2017) on all 17 sensors.

More details on the data collection is available in Section 6.3.5.

Note that combining these datasets is non-trivial as most of them use a different number of markers and varying framerates. The datasets involved in this work are summarized in Table 6.1 . All datasets combined consist of 618 subjects and over 1 million frames of data. To the best of our knowledge no other IMU dataset of this extent is available at the time of writing. We make the following data available for research purposes: the generated synthetic IMU data on AMASS, and the DIP-IMU dataset–including corresponding ground-truth SMPL poses reconstructed from 17 IMUs and the original IMU data.

6.2.4 Deep Inertial Poser (DIP)

Given the training dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^N$ consisting of N training sequences, our task is to learn a function $f : \mathbf{x} \rightarrow \mathbf{y}$ that predicts SMPL pose parameters $\mathbf{y}$ from sparse IMU inputs $\mathbf{x}$ (acceleration and orientation for each sensor). This mapping presents a severely under-constrained problem, since there exist potentially many SMPL poses corresponding to the same IMU inputs. For example, consider the case of knee raises while standing in place. Here the orientation data will remain mostly unchanged and only transient accelerations will be recorded throughout the sequence. This observation led to the use of strong priors and a global optimization formulation in Von Marcard et al. (2017), consisting of orientation, acceleration and anthropometric terms. This approach is computationally expensive and offline, with run-times of several minutes to hours depending on the sequence length. To overcome this limitation, we adopt a data-driven approach and model the mapping with neural networks by using a log-likelihood loss function, implicitly learning the space of valid poses from sequences directly.

Both IMU inputs and corresponding SMPL pose targets are highly structured and exhibit strong correlations due to the articulated nature of human motion. Recurrent neural networks are capable of modeling temporal data and have been previously used in modeling of human motion, typically attempting to predict the next frame of a sequence Fragkiadaki et al. (2015); Martinez et al. (2017); Ghosh et al. (2017). Although we use a different input modality, our model needs to learn similar motion dynamics. In order to exploit temporal coherency in the motion sequence we use recurrent neural networks (RNN) and bi-directional recurrent neural networks (BiRNN) Schuster and Paliwal (1997)—with long short-term memory (LSTM) cells Hochreiter and Schmidhuber (1997).

RNNs summarize the entire motion history via a fixed-length hidden state vector and require the current input $\mathbf{x}_t$ in order to predict the pose vector $\mathbf{y}_t$. While standard RNNs are sufficient in many real-time applications, we experimentally found that having access to both future and past information significantly improves the predicted pose parameters. BiRNNs take all temporal information into account by running two cells in the forward and backward directions, respectively. Compared to RNNs (i.e., unidirectional), BiRNNs exhibit better qualitative and quantitative results by accessing the whole input sequence.

Table 6.1: Dataset overview. "M" denotes MoCap, "I" denotes IMU and "R" RGB imagery. For details on AMASS see Mahmood et al. (2019), for TotalCapture see Trumble et al. (2017). Frame numbers and minutes of AMASS correspond to the number and time length of frames we generated at 60 fps by down-sampling the original data, where required.

Name	Type	Mode	#Frames	#Minutes	#Subjects	#Motions
AMASS	Synth	M	9,730,526	2703	603	11234
TotalCapture	Real	M, I, R	179,176	50	5	46
DIP-IMU	Real	I	330,178	92	10	64

This is in-line with the findings of Von Marcard et al. (2017) where optimizing over the entire sequence was found to be necessary.

Before arriving at the proposed bidirectional architecture, we experimented with simple feed-forward networks and WaveNet van den Oord et al. (2016) variants. We found that these models either perform worse quantitatively or produce unacceptable visual jitter. We assume that RNNs can make better use of the inherent temporal properties of the data and hence produce smoother predictions than non-recurrent variants, especially if they have access to future and past information.

While we train our BiRNN model by using all time-steps, it is important to note that at test time, we use only input sub-sequences consisting of past and future frames in a sliding-window fashion. In our real-time processing pipeline, we only permit a short temporal look ahead to keep the latency penalty minimal. Our evaluations show that using only 5 future frames provides the best compromise between performance and latency. Fig. 6.6 summarizes the impact of window size on the reconstruction quality. We note that in settings with strict low-latency requirements, such as AR, it may be desirable to use no future information at the cost of roughly 1° lower accuracy.

Training with uncertainty

During training, we model target poses y_t with a Normal distribution with diagonal covariance and use a standard log-likelihood loss to train our network:

$$\begin{aligned} \log(p(\mathbf{y})) &= \sum_{t=1}^{T} \log(\mathcal{N}(\mathbf{y}_t|\mu_t, \sigma_t \mathbf{I})), \\ (\mu_t, \sigma_t)_{t=1}^{T} &= f(\mathbf{x}), \end{aligned} \tag{6.2}$$

where f stands for either a unidirectional or bidirectional RNN being trained on sequences of T frames. In other words, our model f outputs μ and σ parameters of a Gaussian distribution at every time step. In-line with the RNN literature, we found that this log-likelihood loss leads to slightly better performance than, for example, mean-squared error (MSE).

Reconstruction of acceleration

The input vector for a single frame, $\mathbf{x}_t = [o_t, a_t]$, contains *orientation* o_t and *acceleration* a_t data as measured by the IMUs. We represent orientations as 3×3 rotation matrices in the SMPL body frame. Before feeding orientations and accelerations into the model, we normalize them w.r.t. the root sensor (cf. Section 6.3.2-6.3.4). This results in 5 input rotation matrices that are all stacked and vectorized into $o_t = \text{vec}(\{\bar{\mathbf{R}}_1^{TB}, \ldots, \bar{\mathbf{R}}_5^{TB}\}) \in \mathbb{R}^{45}$. Similarly, the normalized accelerations are stacked into $a_t \in \mathbb{R}^{15}$.

The acceleration data a_t is inherently noisy and much less stable than the orientations. This issue is further complicated by the subtle differences between real and synthesized accelerations in the training data. In our experiments, we found that different network architectures displayed the tendency to discard most of the acceleration data already at the input level (almost zero weights on the acceleration inputs). For certain motions, the lack of acceleration information causes the model to underestimate flexion and extension of joints. In order to alleviate this problem, we introduce an auxiliary task during training. Our model is asked to reconstruct the input acceleration in addition to pose at training time. This additional loss forces the model to propagate the acceleration information to the upper layers.

Analogous to the main pose task, we model the auxiliary acceleration loss via a Normal distribution with diagonal covariance.

$$\begin{aligned} \log(p(a_t)) &= \sum_{t=1}^{T} \log(\mathcal{N}(a_t | \mu_{a_t}, \sigma_{a_t}\mathbf{I})), \\ (\mu_{a_t}, \sigma_{a_t})_{t=1}^{T} &= f(\mathbf{x} = [o, a]). \end{aligned} \tag{6.3}$$

The pose prediction loss Eq. ((6.2)) and acceleration reconstruction loss Eq. ((6.3)) are complementary to each other and are back-propagated through the architecture with all weights and other network parameters being shared. Only a minimal number of additional trainable network parameters is required to predict μ_{a_t} and σ_{a_t} with sufficient accuracy.

We experimentally show that adding the auxiliary acceleration loss improves pose predictions quantitatively.

Regularization

We train on primarily synthetic data. While the data is sufficiently realistic, slight differences relative to real data are unavoidable. As a consequence, we observed indications of overfitting, and testing on real data yielded less accurate and jerky predictions. To counteract overfitting, we regularize models via dropouts directly on the inputs with a keep probability of 0.8, which randomly filters out 20% of the inputs during training. Randomly masking inputs helps the model to better generalize to the real data and to make smoother temporal predictions.

Fine-tuning with real data

To reduce the gap between real and synthetic data further, we fine-tune the pre-trained models, using the training split of the new dataset (see Sec. 6.3.5). We found fine-tuning particularly effective in situations where specific usage scenarios or motion types were underrepresented in the training data. Hence, this procedure is an effective means of adapting our method to novel situations.

6.3 Implementation Details

6.3.1 Network Architecture

We implemented our network architecture in TensorFlow Abadi et al. (2015). Fig. 6.4 summarizes the architecture details. We used the Adam optimizer Kingma and Ba (2014) with an initial learning rate of 0.001, which is exponentially decayed with a rate of 0.96 and decay step 2000. In order to alleviate the exploding gradient problem, we applied gradient clipping with a norm of 1. We followed the early stopping training scheme by using the validation log-likelihood loss.

6.3.2 Sensors and Calibration

Sensors We use Xsens IMU sensors [2] containing 3-axis accelerometers, gyroscopes and magnetometers; the raw sensor readings are in the sensor-local coordinate frame F^S. Xsens also provides absolute orientation of each sensor relative to a global inertial frame F^I. Specifically, the IMU readings that we use are orientation, provided as a rotation $\mathbf{R}^{IS} : F^S \rightarrow F^I$, which maps from the sensor-local frame to the inertial frame, and acceleration which is provided in local sensor coordinates.

Calibration Before feeding orientation and acceleration to our model, we must transform them to a common body-centric frame, in our case the SMPL body frame F^T. Concretely, we must find the map $\mathbf{R}^{TI} : F^I \rightarrow F^T$, relating the inertial frame to the SMPL body frame. To this end, we place the head sensor onto the head such that the sensor axes align with the SMPL body frame. Consequently, in this configuration, the mapping from head sensor to SMPL frame F^T is the identity. This allows us to set $\mathbf{R}^{TI}$ as the inverse of the orientation $\mathbf{R}_{\text{Head}}$ read from the head sensor at calibration time. All IMU readings can then be expressed in the SMPL body frame:

$$\mathbf{R}_t^{TS} = \mathbf{R}^{TI}\mathbf{R}_t^{IS} = \mathbf{R}_{\text{Head}}^{-1}\mathbf{R}_t^{IS}. \tag{6.4}$$

[2] https://www.xsens.com/

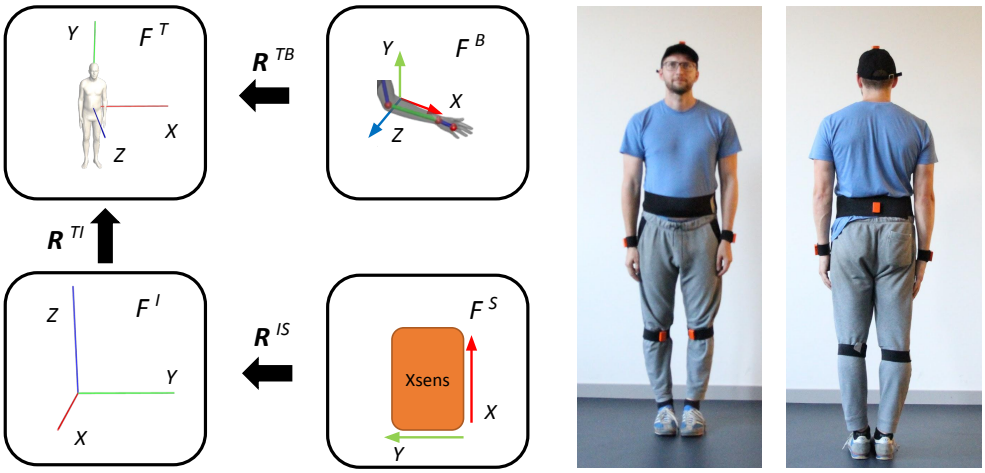


Figure 6.3: Calibration overview. Left: Overview of coordinate frames. Right: Sensor placement and straight pose held by subjects during calibration.

Lastly, due to surrounding body tissue, the sensors are offset from the corresponding bones. We denote this constant offset by $\mathbf{R}^{BS} : F^S \rightarrow F^B$, where F^B is the respective bone coordinate frame. In the first frame of each sequence, each subject stands in a known straight pose with known bone orientation $\mathbf{R}_0^{BT}$, and we compute the per-sensor bone offset as:

$$\mathbf{R}^{BS} = \text{inv}(\mathbf{R}_0^{TB})\mathbf{R}_0^{TS}, \tag{6.5}$$

where $\text{inv}(\cdot)$ denotes matrix inverse.This lets us transform the sensor orientations to obtain virtual bone orientations at every frame

$$\mathbf{R}_t^{TB} = \mathbf{R}_t^{TS}\mathbf{R}^{SB} = \mathbf{R}_t^{TS}\text{inv}(\mathbf{R}^{BS}), \tag{6.6}$$

which we use for training and testing. The interpretation of virtual bone orientations is straightforward: they are the bone orientations as measured by the IMU. The acceleration data is transformed to the SMPL coordinate frame after subtracting gravity, and is denoted as a. Calibration only requires the subject to hold a straight pose for a couple of seconds at the beginning of the recording. Fig. 6.3 provides an overview of the different coordinate frames involved in our calibration process.

6.3.3 Normalization

For better generalization, the input data should be invariant to the facing direction of the person (e.g. a running motion while the subject is facing north or south should produce the same inputs for our learning model). To this end, we normalize all bone orientations with respect to the root sensor, mounted at the base of the user's spine. With $\mathbf{R}_{\text{root}}(t)$ denoting the orientation of the root at time step t, and $\mathbf{R}_s^{TB}(t)$, the orientation of the bone

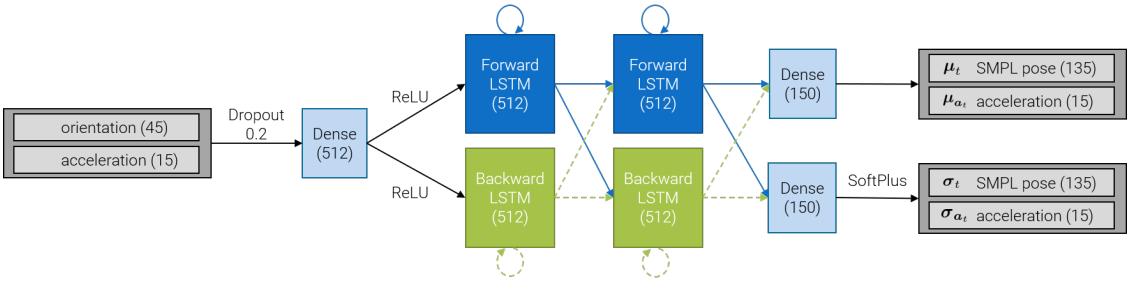


Figure 6.4: Network architecture details. Numbers in brackets are input/output dimensions or number of units in the respective layer. From left to right: The normalized accelerations and orientations are passed to a dense layer before being fed into the two bidirectional recurrent layers as outlined in Fig. 6.2. The output of the last recurrent layer is mapped to two output vectors: the mean SMPL parameters μ_t and mean accelerations μ_{a_t} and the respective standard deviations σ_t and σ_{a_t}. The output of the second dense layer is activated using SoftPlus to enforce non-negativity of the standard deviations.

corresponding to sensor s at time step t, we compute the normalized orientations and accelerations as follows:

$$\bar{\mathbf{R}}_s^{TB}(t) = \mathbf{R}_{\text{root}}^{-1}(t) \cdot \mathbf{R}_s^{TB}(t), \tag{6.7}$$

$$\bar{a}_s(t) = \mathbf{R}_{\text{root}}^{-1}(t) \cdot (a_s(t) - a_{\text{root}}(t)). \tag{6.8}$$

6.3.4 Inputs and Targets

The inputs to DIP are the normalized orientations and accelerations. Using 6 IMUs, the input for one frame,

$$\mathbf{x}_t = [o_t, a_t]^T = [\text{vec}(\bar{\mathbf{R}}_1^{TB}(t), \ldots, \bar{\mathbf{R}}_5^{TB}(t)), \bar{a}_1(t), \ldots, \bar{a}_5(t)]^T,$$

is a vector of dimension $d = (3 \cdot 3 + 3) \cdot 5 = 60$. We experimented with more compact representations of orientation than rotation matrices, such as exponential maps or quaternions; these performed significantly worse than using rotation matrices directly. Rotation matrices elements are bounded between $\{-1, 1\}$ which is good for training neural networks. Similarly for the targets $\mathbf{y}$, instead of regressing to the pose parameters θ of SMPL in axis-angle space, we transform them to rotation matrices and regress them directly. This may seem counter-intuitive because the representation is redundant, but we found empirically that performance is better.

6.3.5 Data Collection

To overcome discrepancies between the sensor data characteristics of the synthetic and real data and to complement the activities portrayed in existing Mocap-based datasets, we recorded an additional dataset of real IMU data, which we call DIP-IMU.

We recorded data from 10 subjects (9 male, 1 female) wearing 17 Xsens sensors (see Fig. 6.3). All the subjects gave informed consent to share their IMU data. To attain ground-truth, we ran SIP Von Marcard et al. (2017) on all 17 sensors. To compensate for different magnetic offsets across IMUs, a heading reset is performed first. The sensors are aligned in a known spatial configuration, after which the heading is reset. Subsequently, the sensors are mounted on the subject and the calibration procedure (cf. Section 6.3.2) is performed. Participants were then asked to repeatedly carry out motions in five different categories, including controlled motion of the extremities (arms, legs), locomotion, and also more natural full-body activities (e.g., jumping jacks, boxing) and interaction tasks with everyday objects. In total, we captured approximately 90 minutes of additional data resulting in the largest dataset of real IMU data (Table 6.2).

Table 6.2: Dataset capture protocol used to record DIP-IMU.

Categories	Motions (# Repetitions)	# Frames	Minutes
Upper Body	Arm raises, stretches, and swings (10). Arm crossings on torso and behind head (10).	116,817	32.45
Lower Body	Leg raises (10). Squats (shoulder-width and wide) (5). Lunges (5).	70,743	19.65
Locomotion	Walking straight (3). Walking in circle (2). Sidesteps, crossing legs (1). Sidesteps, touching feet (1).	73,935	20.54
Freestyle	The subject can select one of the following activities: jumping jacks, tennis, kicking/boxing, push-ups, basketball. Choice of jumping jacks is predominant.	18,587	5.16
Interaction	The subject sits at a table and interacts with everyday objects, such as keyboard, mobile device, toys, grabbing objects in front of them, touching mounted screen displays. Freestyle for 1 minute.	50,096	13.92

6.4 Experiments

To assess the proposed method, we performed a variety of quantitative and qualitative evaluations. We compare our method to the offline baselines SIP Von Marcard et al.

(2017) and SOP (reduced version of SIP not leveraging accelerations), and perform self-comparisons between the variants of our architecture. Here we distinguish between two distinct settings. First, we compare performance in the offline setting. That is, we use test sequences from existing real datasets (TotalCapture and DIP-IMU). Second, one of the main contributions of our work is the real-time (online) capability of our system. To demonstrate this, we implemented an end-to-end live system, taking IMU data as input and predicting SMPL parameters as output.

6.4.1 Quantitative Evaluation

In this section we show how our approach performs on several test datasets. We report both mean joint angle error, computed as in Von Marcard et al. (2017), and positional error.

Offline evaluation

First, we report results from the offline setting, in which all models have access to the whole sequence. This setting is a fair comparison between our model and the baselines since SIP and SOP solve an optimization problem over the whole sequence. Table 6.3 summarizes the results with our models performing close to or better than the SIP baseline. The best configuration (BiRNN (Acc+Dropout)) outperforms SIP by more than one degree. Fig. 6.5 (left) shows the angular error distribution over the entire TotalCapture dataset. The peak is around 8° error.

The combination of dropouts on the inputs and use of the acceleration loss improve both RNN and BiRNN models. Note that, due to its access to the future steps, BiRNNs perform qualitatively better and produce smoother predictions than the uni-directional RNNs.

Fine-tuning on real data

While the techniques shown in the previous section perform reasonably well on TotalCapture, a significant performance drop is evident on DIP-IMU. This is due to the difference in motions in our new dataset and the aforementioned gap between real and synthetic data distributions. However, the results we have analyzed so far stem from models trained without access to the DIP-IMU data and hence have not seen the types of poses and motions contained in DIP-IMU. We now report the results from our best configurations after fine-tuning on the DIP-IMU data. We fine tune the network on the training split and test it on the held-out set. Tables 6.3 and 6.4 show results from the offline and online setting. We find a clear performance increase on DIP-IMU, which is now comparable to TotalCapture. This is further illustrated by the error histogram on DIP-IMU before and after fine-tuning (cf. Fig. 6.5). Note that the performance on

Table 6.3: Offline evaluation of SOP, SIP, RNN and BiRNN models on TotalCapture Trumble et al. (2017) and DIP-IMU. Errors reported as joint angle errors in degrees and positional errors in centimeters. Models with *Dropout* (D) are trained by applying dropout on input sequences. *A* corresponds to acceleration reconstruction loss. SOP, SIP and BiRNN have access to the whole input sequence while RNN models only use inputs from the past. BiRNN^F $(after fine-tuning)$ $isBiRNN(Acc+Dropout)fined-tunedonDIP-IMUusingaccelerationreconstructionlossanddropoutaswell.$

	TotalCapture				DIP-IMU			
	μ_{ang}[deg]	σ_{ang}[deg]	μ_{pos}[cm]	σ_{pos}[cm]	μ_{ang}[deg]	σ_{ang}[deg]	μ_{pos}[cm]	σ_{pos}[cm]
SOP	22.18	17.34	8.39	7.57	27.78	19.50	8.23	6.74
SIP	16.98	13.26	5.97	5.50	24.00	16.91	6.34	5.86
RNN^D	16.83	13.41	6.27	6.32	35.66	19.96	13.38	8.84
RNN^A	16.07	13.16	6.06	6.01	41.00	29.36	15.30	12.96
RNN^{AD}	16.08	13.46	6.21	6.27	30.90	18.66	11.84	8.59
BiRNN^D	15.86	13.12	6.09	6.01	34.55	19.62	12.85	8.62
BiRNN^A	16.31	12.28	**5.78**	5.62	37.88	24.68	14.31	11.30
BiRNN^{AD}	**15.85**	12.87	5.98	6.03	31.70	17.30	12.07	8.72
BiRNN^F	16.84	13.22	6.51	6.17	**17.54**	11.54	**6.49**	5.36

TotalCapture decreases only minimally, indicating that no catastrophic forgetting takes place.

Online evaluation

Next, we select our best performing configuration and evaluate it in the online setting. We do not evaluate performance of SIP and SOP since these baselines can not run online. Note that now the RNN configurations no longer have access to the entire sequence, but only to past frames in the case of the uni-directional RNN, and to a sliding window of past and a few future frames in the case of the BiRNN. Table 6.4 indicates that the networks obtain good pose accuracy in the online setting. Notably, the BiRNN with access to 50 past frames even slightly outperforms the offline setting on TotalCapture. This may be due to the accumulation of error in the hidden state of the RNN and the stochasticity of human motion over longer time spans.

In the online setting, the influence of the acceleration loss is most evident. If evaluated on 20 past and 5 future frames on TotalCapture, a BiRNN without acceleration loss performs worse ($16.26^\circ \pm 13.54^\circ$) than one using the acceleration loss ($15.88^\circ \pm 13.57^\circ$). For 50 past and 5 future frames the error increases to $16.10^\circ \pm 13.42^\circ$ (compared to $15.77^\circ \pm 13.41^\circ$).

Table 6.4: Online evaluation of BiRNN models on TotalCapture Trumble et al. (2017) and DIP-IMU. We select the best performing model from our offline evaluation, i.e., (Acc+Dropout). Numbers in brackets (x,y) mean that this model is evaluated in online mode using x past and y future frames. (fine-tuning) means that the model was fine-tuned on DIP-IMU. The symbols "d" and "c" mean units "degree" and "cm" respectively.

	TotalCapture				DIP-IMU			
	μ_{ang}[d]	σ_{ang}[d]	μ_{pos}[c]	σ_{pos}[c]	μ_{ang}[d]	σ_{ang}[d]	μ_{pos}[c]	σ_{pos}[c]
BiRNN$^{(20,5)}$	15.88	13.57	6.00	6.16	38.42	25.06	14.49	11.42
BiRNN$^{(50,5)}$	**15.77**	13.41	**5.96**	6.13	39.11	24.70	14.81	11.52
BiRNN^{F}	16.84	13.22	6.51	6.17	**17.54**	11.54	**6.49**	5.36
BiRNN$^{(20,5),F}$	16.90	13.83	6.46	6.26	18.49	12.88	6.63	5.54
BiRNN$^{(50,5),F}$	16.74	13.64	6.42	6.22	18.14	12.75	6.52	5.48

Influence of future window length

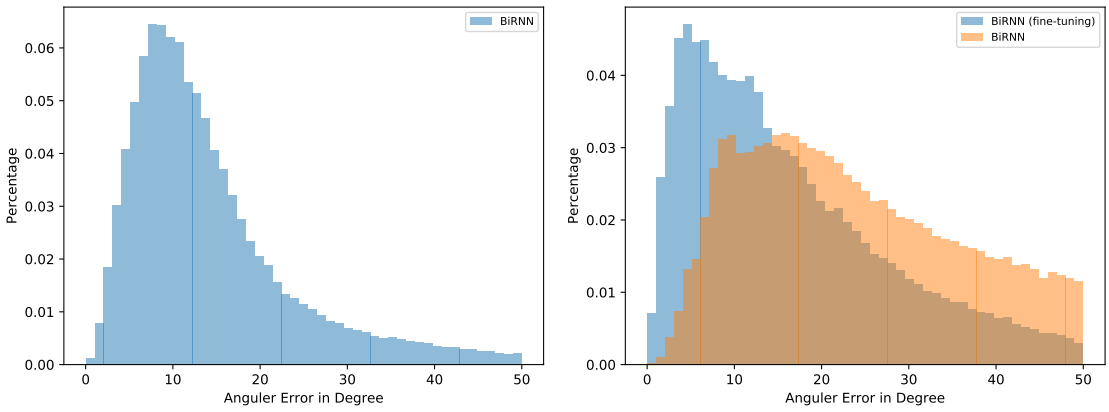
Our final implementation leverages a BiRNN to learn motion dynamics from the data. At training time, the entire sequence is processed, but at runtime, only a subset of frames is made available to the network. Fig. 6.6 summarizes the performance of the network as function of how many frames of past and future information are available at test time. We experimentally found that using 5 future and 20 past frames is the best compromise between prediction accuracy and latency penalty; we use this setting in our live system.

6.4.2 Qualitative Evaluation

We now further assess our approach via qualitative evaluations. Based on the above quantitative results, we only report results from our best model (BiRNN). We use the model in offline mode to produce the results in this section. Section 6.4.3 discusses the results when using it in online mode. Please see the accompanying video at: `https://www.youtube.com/watch?v=p1fmpOWA504`.

Playground (real)

First, we compare to SIP and SOP on the Playground dataset Von Marcard et al. (2017). Playground is challenging because it is captured outdoors and contains uncommon poses and motions. Since the dataset contains no ground truth, we provide only qualitative results. Fig. 6.7 shows selected frames from a sequence where the subject climbs over an obstacle. We find that SOP has a lot of trouble in reconstructing the leg motion and systematically underestimates arm and knee bending. Our results are comparable to SIP although sometimes the limbs are more outstretched than in the baseline. However, note that SIP optimizes over the whole sequence and is hence computationally very expensive, whereas ours produces predictions in milliseconds.



(a) b (b) b

Figure 6.5: Histogram of joint angle errors (°). Left: Error distribution on TotalCapture with the offline BiRNN model. Right: Performance on DIP-IMU before and after fine-tuning as described in Section 6.4.1.

TotalCapture (real)

Here we provide a qualitative comparison of our method and the baseline on the TotalCapture dataset Trumble et al. (2017). Fig. 6.8 summarizes three different sample frames from the dataset. We note that for challenging motions such as back bending (bottom row) and leg raises (middle row), our model outperforms both SIP and SOP and is very close to the reference. Fig. 6.8 also shows a case where our model successfully reconstructs a leg-raise when SIP and SOP fail. Note however, that this difficult motion also fails to be reconstructed by our model at times (cf. Section 6.5.2).

DIP-IMU (real)

Lastly, we illustrate results on our own DIP-IMU dataset (cf. Sec. 6.2.3 and Table 6.2). Here the reference ("ground truth") is obtained by running SIP using all 17 IMUs. At test time, however, we only use 6 IMUs as input for our method, and for the baselines SIP and SOP. Fig. 6.9 summarizes several sequences from the dataset. It is evident that our model outperforms both SIP and SOP qualitatively in a consistent way. We see that SIP/SOP creates a lot of inter-penetrations between limbs and the torso. Our model more faithfully reproduces the arm motions over a large range of frames and poses. Interestingly, our model rarely produces inter-penetrations and produces smooth motion despite noise in the inputs (see video at `https://www.youtube.com/watch?v=p1fmpOWA504`) and without any explicit smoothness or inter-penetration constraints. Hence, DIP learns a mapping from IMU data to the space of valid poses and motion. Smoothness may be explained by the regularization of training via dropouts.

# past frames \ # future frames	0	1	5	10	20	50	100	200
200	16.447	16.108	15.761	15.668	15.649	15.692	15.733	15.752
100	16.417	16.082	15.750	15.652	15.622	15.646	15.675	15.690
50	16.424	16.084	15.768	15.662	15.615	15.619	15.630	15.638
20	16.510	16.177	15.880	15.759	15.684	15.642	15.634	15.622
10	16.578	16.240	15.944	15.815	15.721	15.658	15.638	15.614
5	16.695	16.337	16.009	15.870	15.766	15.690	15.663	15.632
1	17.221	16.683	16.168	15.997	15.883	15.815	15.794	15.768
0	18.008	17.138	16.530	16.346	16.211	16.154	16.150	16.142

Figure 6.6: Performance of BiRNN as function of past and future frames on TotalCapture. Numbers are mean joint angle error in degrees. Zero frames means no frames contribute to the prediction from the past or future respectively, i.e. only use the current frame.

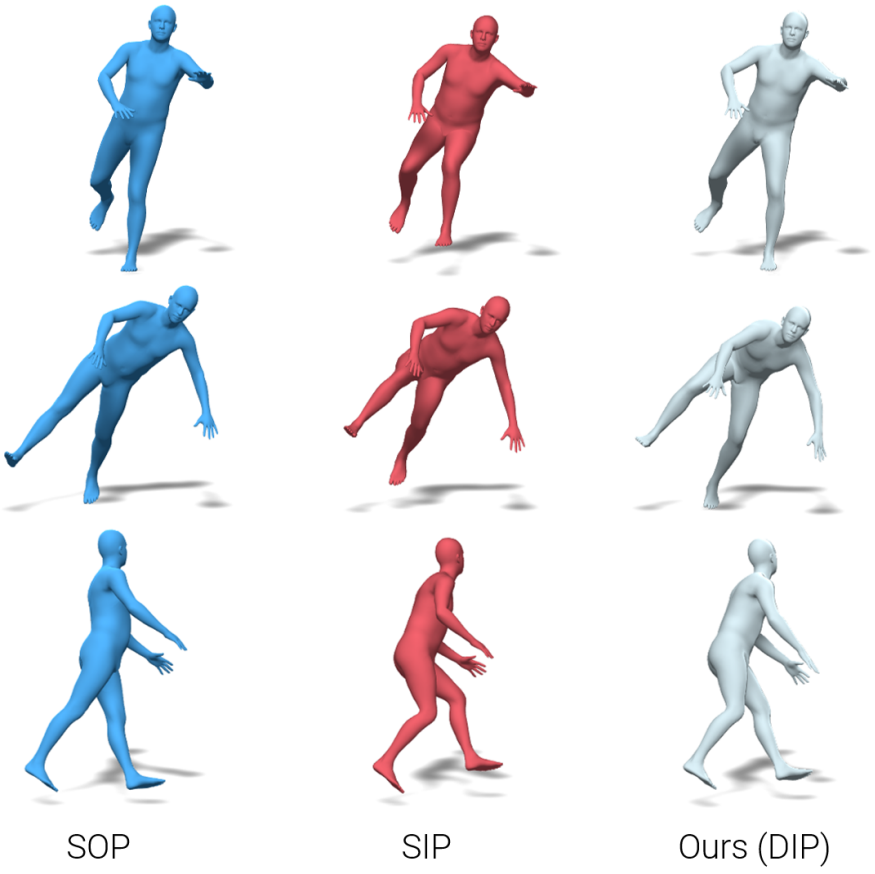


Figure 6.7: Selected frames from Playground dataset.

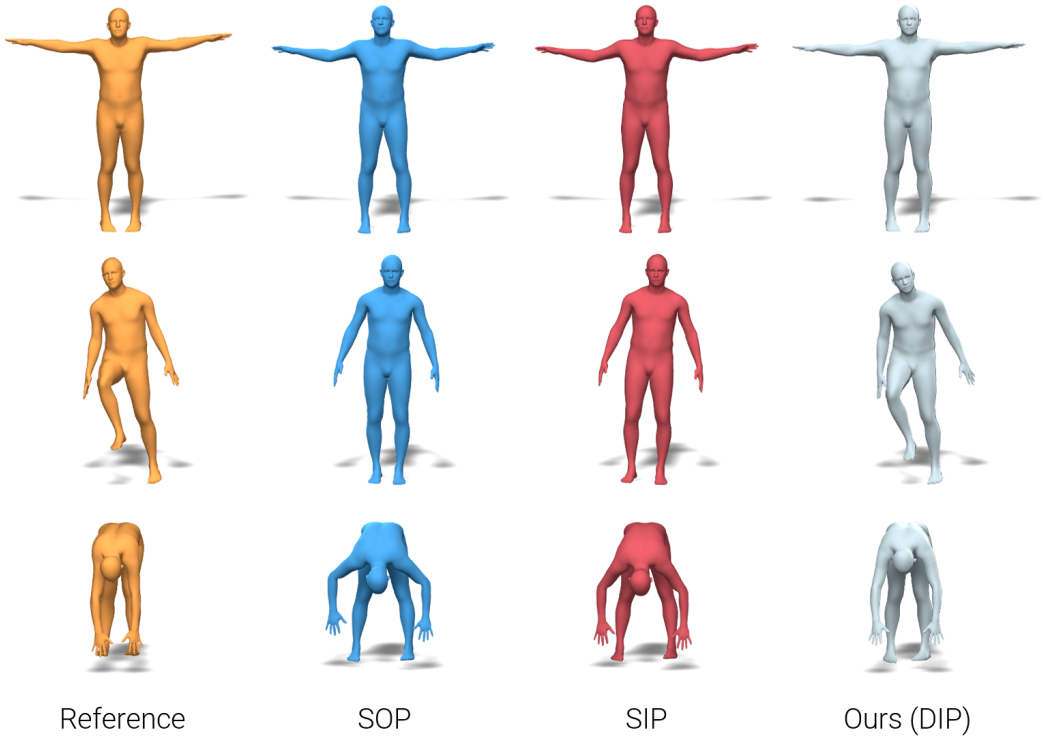


Figure 6.8: Sample frames from TotalCapture data set (S1, ROM1).

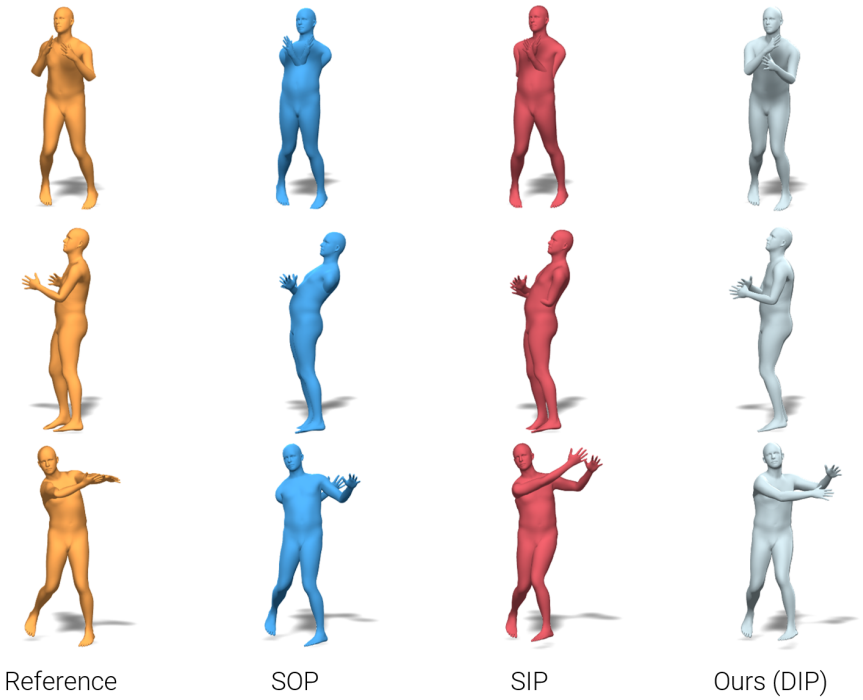


Figure 6.9: Sample frames from DIP-IMU (S10, Motion4).

6.4.3 Live Demo

To demonstrate that our model runs in real-time, we implemented an on-line system that streams the sensor data directly into the model and displays the output poses. The raw IMU readings are retrieved via the Xsens SDK, the model's output is displayed via Unity and all communication is performed via the network. Example frames from the live demo are shown in Fig. 6.1 and Fig. 6.10. The results are best viewed in the supplementary video. For the live demo we use the online version of the fine-tuned BiRNN model as explained in Section 5.2, i.e. using 20 frames from the past and 5 from the future. The system runs at approximately 29 fps while still producing faithful results.

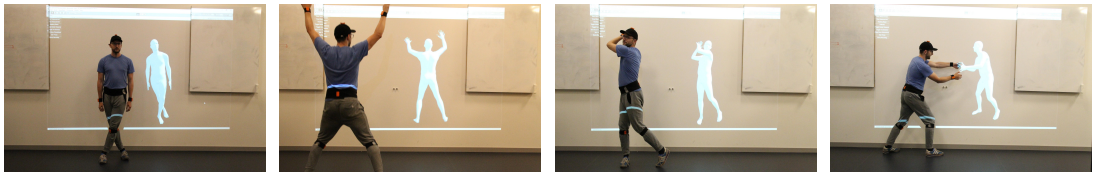


Figure 6.10: Sample frames from the live demo showing that our model is able to handle various motion types. See also Fig. 6.1.

6.5 Discussion and Limitations

6.5.1 Generalization

In this chapter we have shown that our model is able to generalize well to unseen data based on the following observations. (i) We achieve good results on a held-out dataset with real IMU recordings (Total Capture) despite training on synthetic data only (cf. Section 6.4.1). (ii) We show good qualitative performance on another held-out dataset, Playground, and in the live demo (cf. Section 6.4.2 and 6.4.3). This is challenging to achieve due to differences in motions, sensors, and data preprocessing across the datasets. (iii) The system is robust w.r.t. different root orientations (cf. Fig. 6.1). However, robustness to even more poses, datasets and settings is still the subject of future work. We hypothesize that one of the main limitations is the difficulty of modeling accelerations (synthetic and real) effectively. This is the main reason for fine-tuning on DIP-IMU, which improves generalization but certainly does not have to be the final solution to this problem. In the following we report additional experiments to provide insight into these issues.

Synthetic vs. real We first train a BiRNN on a smaller, real dataset (DIP-IMU) as opposed to a large, synthetic one (AMASS). We subsequently evaluate this model on TotalCapture, where we notice a drop in performance of around 5.2° (21.03° ±16.35°). Testing on the DIP-IMU held-out set yields an error of (18.84° ±14.08°), which is comparable to the performance when training with synthetic and fine-tuning with DIP-IMU (17.54°). However, the latter yields better performance on TotalCapture. These results illustrate the benefits of a large synthetic motion database.

Re-Synthesis To analyze the impact of differences between synthetic and real data, we synthesize the IMU measurements for the real datasets (TotalCapture and DIP-IMU). We then evaluate our best BiRNN on these synthetic versions of TotalCapture and DIP-IMU and compare it with the performance on real data. The model performs better on the synthetic version of both TotalCapture (improvement by 2.71° to 13.14° ±10.50°) and DIP-IMU (improvement by 8.84° to 22.86° ±15.70°), highlighting domain differences that need to be addressed. In summary, we hypothesize that differences in accelerations lie at the core of this problem.

6.5.2 Failure Cases

Fig. 6.11 (top) shows typical failure cases, taken from an example sequence in TotalCapture. While the model is robust to various root orientations in the live demo, extreme cases, where the body is parallel to the floor (such as push-ups), are challenging. Finally, we compute the worst 5% poses on the test set of DIP-IMU, which results in a mean joint angle error of 43.68° ±8.53° degrees.

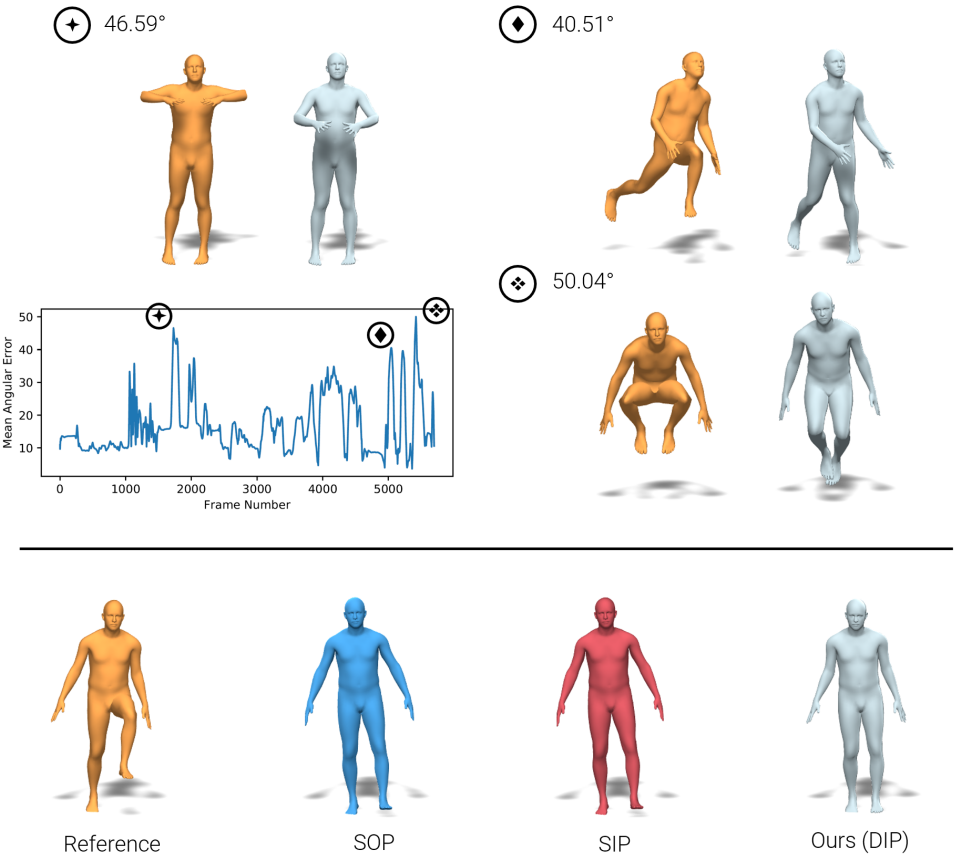


Figure 6.11: Top: Per-frame average angular error over a sequence from TotalCapture (S3, ROM1) including 3 maximum error poses. The mean angular error for this sequence is $16.73° \pm 8.55°$. Bottom: Typical failure case from TotalCapture. Our model and both baselines fail to reconstruct the leg raise.

Leg raises are among the most difficult motions as the sensors show very similar orientation readings while performing this motion. Hence, only acceleration can disambiguate these Von Marcard et al. (2017). However, sensitivity to IMU placement, environmental factors, and different noise characteristics per sensor make it extremely challenging to integrate them effectively into a learning-based approach. Fig. 6.11 (bottom) shows how both our model and the baselines fail to reconstruct a leg raise. We believe that these problems arise from the fact that our model struggles to fully exploit the acceleration information. Hence, future work should focus on addressing this challenge by modeling the noise in acceleration and exploring new sensor placement.

6.6 Conclusion and Discussion

We presented a new learning-based pose-estimation method that requires only 6 IMUs as input, runs in real-time, and avoids the direct line-of-sight requirements of camera-based systems. We leverage a large Mocap corpus to synthesize IMU data (orientation and acceleration) using the SMPL body model. From this synthetic data, we show how to learn a model (DIP) that generalizes to real IMU data, obtaining an accuracy of 15.85° angular error on TotalCapture. We exploit temporal information by using a bi-directional RNN that propagates information forward and backwards in time; at training time DIP has access to full sequences, whereas at test time the model has access to the last 20 frames and only 5 frames in the future. This produces accurate pose estimates at a latency of only 85ms. Even satisfying the real-time requirement, DIP performs comparably to, or better than, the competing off-line approach, SIP. Furthermore, DIP produces results that are smooth and generally without inter-penetrations. This demonstrates that DIP learns a mapping to the space of valid human poses without requiring explicit smoothness or joint angle limits.

Future work should address capturing multi-person interactions and people interacting with objects and the environment. While the focus of this work has been a system based purely on wearable sensors, some applications admit external, or body mounted cameras Rhodin et al. (2016a). It would be interesting to integrate visual input with our tracker in order to obtain even better pose estimates, especially to capture contact points, knee bends and sitting-down poses, which are difficult to recover using only 6 IMUs. While our approach runs in real-time, transferring the motion data over the Internet may introduce latency, which is a problem for virtual social interaction. Hence, we will explore ways to predict into the future to reduce latency. Finally, unlike Von Marcard et al. (2017), no global translation is considered in our method. This limitation can be critical in some application scenarios. We see two possible solutions to this. First, a GPS signal, which is integrated into most phones, could be integrated into DIP to obtain reasonable global position. Another potential way is to regress the global translations directly from the temporal IMU inputs. We leave this for future work.

We have demonstrated the capabilities of DIP by displaying its pose predictions in real time. We believe that real-time pose estimation methods, which require only a small number of wearable sensors like DIP, will play a key role for emerging interactive technologies such as VR and AR.

Chapter 7

Activity Estimation from One Smartwatch

7.1 Introduction

After more than half one century of development since the initial introduction of the concept, Artificial Intelligence (AI) has finally embraced an unprecedented matureness and practicalness, seeing numerous applications in almost every aspect of our daily life[1] (Langton, 1997). Common services we are using on an everyday basis include: Google[2] in a search engine for personalized information retrieval; Amazon[3] in online retail for accurate product recommendation; Uber[4] in shared transportation, for collaborative and optimized path planning; MobileEye[5] in autonomous driving, in the current stage for road condition sensing and driver assistance, with the ultimate goal of achieving fully automatic self-driving vehicles; FaceID[6] in automatic face recognition, for highly secure identity identification; SnapChat[7] in enhanced instant messaging, for virtual makeup and other user experiences. The list keeps growing. Though focusing on different aspects of human life, all these applications merge with and benefit our way of living remarkably, showing or even partially achieving the great potential of reshaping how the world works and how we think about intelligent agents (Wooldridge and Jennings, 1995).

Arguably quite a large portion of all these diverse applications are based on Deep Learning (DL) (LeCun et al., 2015) and its various variants like Convolutional-Neural-Networks (CNNs) (Krizhevsky et al., 2012), LSTM (Hochreiter and Schmidhuber, 1997), Reinforcement Learning (Sutton and Barto, 2018), and more recent Generative-Neural-Networks (GANs) (Goodfellow et al., 2014), the biggest breakthrough in the area of AI for the last decades. It is shown that a powerful DL model, together with a huge

[1] The work presented in this chapter has not yet been published.
[2] https://www.google.com
[3] https://www.amazon.com
[4] https://www.uber.com
[5] https://www.mobileye.com/
[6] https://www.apple.com/lae/iphone-xs/face-id/
[7] https://www.snapchat.com/

collection of data samples drawn from the domain of interest, usually in the magnitude of millions, is capable of summing up intrinsic knowledge about the problem at hand, then generalizing pretty well to unseen samples. Here the data corpus used to feed the model is as important as, if not more important than, the DL model under the hood. Firstly emerging in the area of Computer Vision, this paradigm is partially stimulated and boosted by the construction of many big dataset like Sintel (Dosovitskiy et al., 2015) for optical flow, FAUST (Bogo et al., 2014) for human body geometry, ShapeNet (Chang et al., 2015) for the geometry and texture of general objects in 3D, MSCOCO (Lin et al., 2014) for object detection and segmentation, LFW face (Learned-Miller et al., 2016) for face detection in the wild, ImageNet (Deng et al., 2009) for general image recognition. This philosophy is adopted in other closely related branches of AI like Natural Language Processing (NLP) (Manning et al., 1999; LeCun et al., 2015) and Speech Recognition (Hinton et al., 2012).

However, the advancement of technology is always a sword of two edges. The core potential threat is whether these powerful tools we create ourselves will be used against us or in an unexpected and negative way. Some people are already worried that the increasingly intelligent artificial intelligence systems might overturn the human race one day, making the classic scenes in countless sci-fiction novels and movies a reality. Though a fully complete human-like intelligent algorithm, so-called Universal Artificial Intelligence (UAI) (Hewitt et al., 1973), seems not possible in the short term, the current AI systems are starting to make workers in specific job sectors deeply concerned about their job security. Taxi-driving and language translation probably are among the top endangered professions. The threat is even more severe if we narrow down the scope to creating and editing facial images and videos enabled by state-of-the-art Computer Vision technologies. This type of systems, collectively named *DeepFakes* (Schwartz, 2018; Boylan, 2018), can synthesize different kinds of face images or videos that do not exist in reality but are hardly distinguishable from the real ones. The Face2Face system (Thies et al., 2016) shows for the first time the realistic expression transfer, requiring only an ordinary RGB camera to capture the expression of the source actor and a video depicting the face expression of the target actor. The following alternatives like paGAN (Nagano et al., 2018) and Warp-Guided GANs (Geng et al., 2018) advance the realism to a higher level. If these systems demonstrate the unlimited potential of this type of "surreal-face-swapping" technology, then the DeepNude[8] APP that renders chosen celebrities naked exemplifies how they can also be used in a negative way.

Until now, the public's primary concern about the privacy and security issue introduced by AI is mostly constrained in special devices like cameras, voice recorders, or 3D scanners. The general public seems to hold the opinion that one can stay free from these threats as long as he or she is away from the mentioned sensitive sources. One natural question to ask about it is whether this assumption is true or not. Unfortunately, our investigation of this question answers quite differently. With one single wrist-worn

[8]https://www.deepnude.com/

IMU shipped with consumer-level smartwatches, we show that it is possible to estimate which typical motions (15 classes in our current setting) the actor is doing at an accuracy way higher than random guess (74% vs 7%). This not only opens the door to an affordable, practical, ubiquitous, and lasting solution to automatic HAR but also incurs more thoughts about the security and privacy issues. The effect can be profound and far-reaching. The user might not only want to think twice before they wear their smartwatches, but also they want to double-check the various intelligent devices in their rooms, including but not limited to smart TVs and intelligent speakers.

Our angle towards personal information acquisition from smartwatches containing one IMU is quite different from the previous work (Vertens et al., 2015; Albright et al., 2011; Khedr and El-Sheimy, 2017). It is well-known that smartwatches can detect different kinds of personal health and motion information like heart rate, step count, and even sleeping time. These signals are fine-grained, lack semantic meaning, and do not indicate high-level personal motion and activity. Here we aim to infer what the person is doing daily in a non-invasive way, using one smartwatch. Our methodology is purely data-driven, with Deep Learning enabled Multi-Layer-Perceptrons (MLP) as the underlying model. Our work also deviates from the IMU-based HAR approaches, where constrained activities are considered, and usually, more than one commercial IMU are assumed (Chen et al., 2006; Bayati et al., 2011; Reiss and Stricker, 2012; Jordao et al., 2018). These works usually report a quite high accuracy, even close to 90%, but the setting is limited, the dataset lacks variation in activities, and the scenarios of consideration usually are far away from real life. Instead, we focus on the most common human motions in different scenarios like indoors, playgrounds, supermarkets and, offices. We believe our study shows the potential of expanding the application scope of IMU-based HAR, making it more practical, and also reveals more about the hidden privacy and security issue easy to be ignored and underestimated by the public.

7.2 Data Recording

To enable the research, we managed to gather a large-scale, diverse and realistic IMU dataset recorded when the subjects perform daily activities. Here we delicately choose Apple Watch[9] as the hardware to use since it is popular nowadays and also provides a mature software development environment. Once activated, one specific APP runs on the watch and extracts the IMU readings. See Figure 7.1 for the user interface of the APP. We focus on 15 everyday daily activities like running, walking, shopping, and sleeping for the current study. Note that these typical motions cover various indoor, outdoor, playground, and supermarket scenarios. More details are revealed in the following sections.

[9] https://www.apple.com/watch/

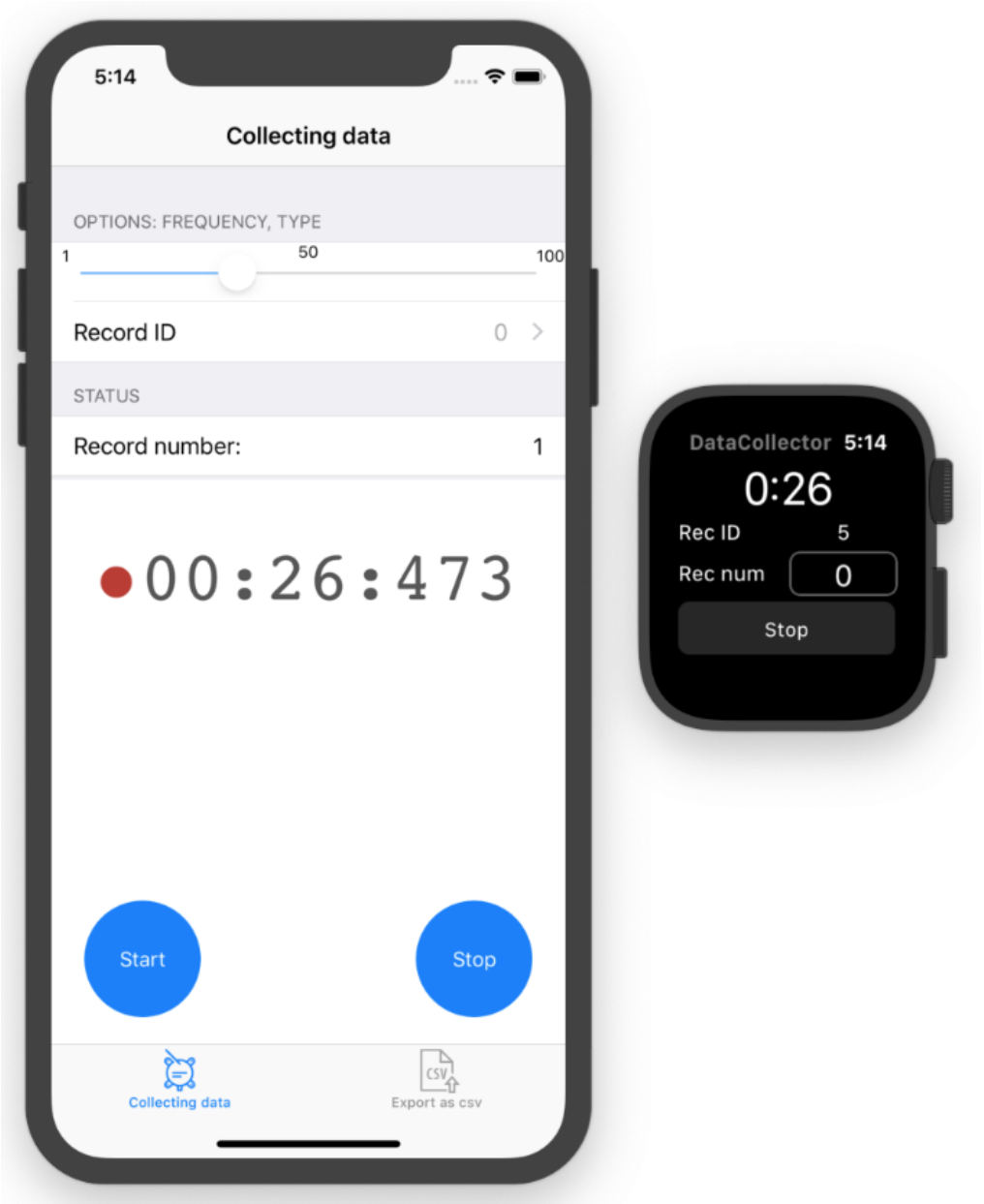


Figure 7.1: User interface of the WatchOS APP developed and used in the project.

7.2.1 Hardware Platform

There are many kinds of smartwatches available on the market, with the prices ranging from less than one hundred to more than one thousand. The difference in price mainly comes from the additional features the watches carry. Some common ones include fancier design, more sensors, and better hardware support. For us the gadget of the most interest is the IMU, a standard and cheap accessory nowadays. Among all these different options, we choose the Apple Watch Series 4[10], the latest edition of the series, purely for its popularity among the public and its mature and well-received software development environment. Please note that the data capture procedure can be easily performed with other alternative devices as long as they contain a decent IMU, and the study we conduct on the data is independent of the hardware used.

7.2.2 IMU Recorder

The recording APP conforms with the WatchOS development framework and is composed of two parts: the one running on the watch to extract the IMU measurement on the fly; the other runs on the paired iPhone to relay the captured data and upload it to the cloud. The current version supports both per-frame transfer and all-frames as a whole transfer. There is no apparent difference between these two when the recording time is limited to roughly 20 minutes, so we adopted the all-frames as a whole transfer mode for its lower battery consumption feature. The communication between a smartwatch and a mobile phone is enabled by bluetooth, which suffers from package missing sometimes, but is still good enough for our purposes.

The module on the watch (right in Figure 7.1) purely serves as the switch. The actor presses the "Start" and "Stop" buttons to activate and stop the data capture session. In our current setting, where all-frames as a whole transfer mode is utilized, the IMU readings are firstly accumulated in the internal storage of the watch and then passed to the iPhone once the session is done. After each session, the user is prompted to upload the gathered data to the cloud via the GUI on the mobile phone screen (left in Figure 7.1).

7.2.3 Dataset Acquisition

We recruit 11 persons to construct the new dataset. Two of the subjects are females, while the remaining ones are males. Their average age is around twenty-seven years old, and their nationalities are diverse, spanning countries like India, Germany, and China. For the current study, we mainly focus on everyday activities. We collect the motions the actors frequently perform in daily life, then pick the 15 ones that get the most votes from the participants. The tags of these motions are shown in Table 7.1. We dub this dataset DHIRI (Daily Human motIon measuRed in Imu).

[10]https://www.apple.com/apple-watch-series-4/

Walk	Run	Sleep	Read	Brush_teeth
Wash_hands	Open_door	Sweep	Shop	Type_keyboard
Drink	Eat	Talk / Converse	Make_phone_call	Watch_tv

Table 7.1: The 15 daily activities we consider in this study. They cover the common indoor and outdoor scenarios like living room, office, restaurant, and playground. Note these motions occupy behaviors shared across people, thus being quite representative.

Figure 7.2: 11 subjects take part into the data capture in our study. Shown here are some illustrative frames extracted from the accompaning videos used to label the motion. Note the variation in the recording place, and also the clock over the top-right of the images, which is used to synchronize the video with IMU readings.

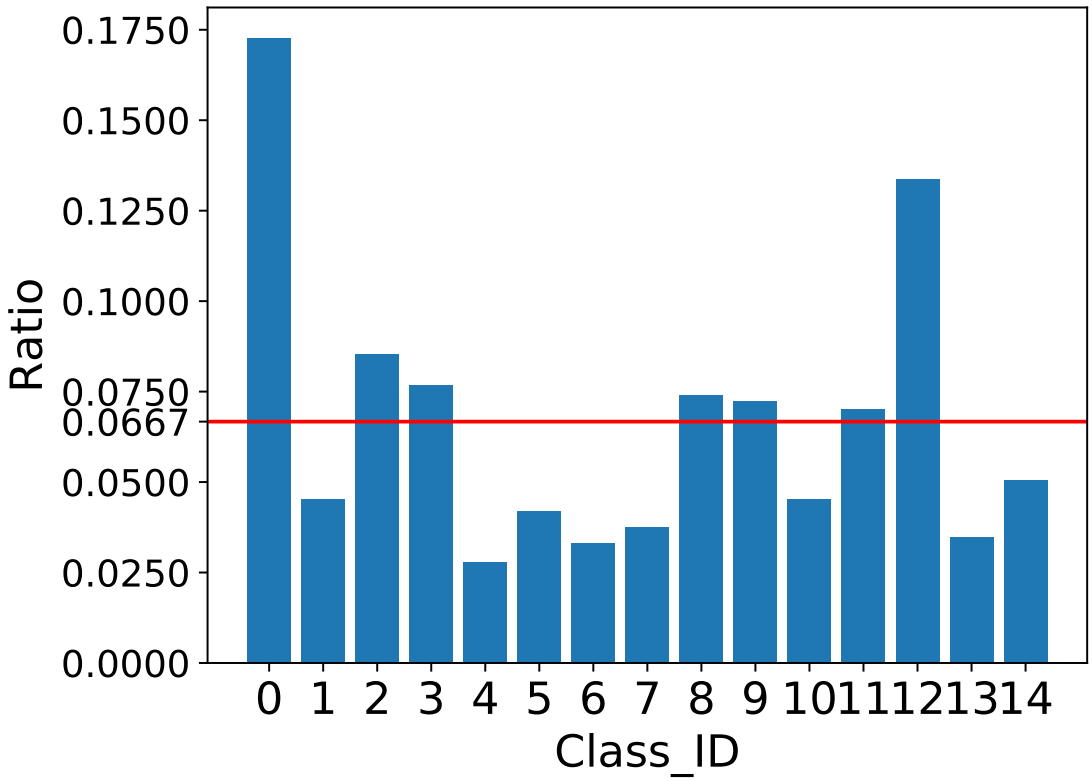


Figure 7.3: Distribution of the 15 classes in the captured dataset. The red line indicates the average ratio, namely 1.0 / 15.

The capture happens in two phases. Firstly the subjects are asked to perform every motion one by one discretely, each lasting for roughly 5 minutes. The subjects are encouraged to show as many different ways of doing the same kind of motion as possible, thus making the captured data rich in various motion patterns. Then in the second phase, the participants are free to alternate among these motions in any way they like, as long as it is natural and continuous, for 3 to 4 times, each one being around 10 minutes. The frame rate of IMU reading was fixed as 50 FPS over the whole process. Altogether we collected around 2.3 million frames, or equivalently 18 hours, making DHIRI one of the largest real IMU datasets. Four of the subjects doing four different kinds of motions are depicted in Figure 7.2. We find the classes of this dataset are not so well balanced, with more than half of them below the average ratio (see Figure 7.3). The first three most frequent activities are Walk, Talk/Converse, and Sleep, while the three least frequent ones are Brush_teeth, Open_door, and Make_phone_call.

Compared with existing datasets, the new one we build features several appealing attributes. Firstly by design, it focuses on typical motions exhibited by ordinary people in daily life, thus more suitable for studies targeting the general public. This is in contrast

to previous benchmarks limited to special places like a factory. Secondly, our dataset is more realistic. In some previous work, the participants are required to aid the data labeling work, mainly through a carefully-designed APP user interface, when the capture is going on. Though convenient, this way of collecting ground-truth labels can potentially intervene with the data capture process. We avoid this problem by explicitly asking the actors to make the specific motions in the first stage and using another mobile phone to record the footage for later manual annotation in the second stage. The participants receive no further guidance once the capture session starts. We believe this mechanism results in more natural data.

7.3 Methodology

Following the previous IMU-enabled HAR work, we also formulate the problem as a classification problem. That is, for every frame of IMU readings, we need to assign to it a discrete activity label. This problem is intrinsically heavily ambiguous and complex since the linear acceleration, and angular rate signals from one single wrist-worn IMU only carry limited information about how the wrist moves, which per se weakly correlates with whole-body motion. A similar observation is also made in the Deep Inertial Poser paper (Huang et al., 2018) where several sparse IMUs are used to infer the full-body pose. There, 50 past frames and five future frames are combined to better constrain the pose better to estimate, while one LSTM is explicitly adopted to propagate contextual information. Here similarly, we also concatenate IMU readings from several continuous frames as the input unit. Please note that different from (Huang et al., 2018) where the absolute orientation of body parts is known, we only have the instantaneous measurement that reads how fast the attached body part moves.

The overview of the whole pipeline is depicted in Figure 7.4. The input to our system is always a collection of temporal IMU sequences sampled at 10 FPS, each one in the form of $\{(a_0, r_0), (a_1, r_1), \ldots, (a_N, r_N)\}$, with corresponding per-frame ground-truth activity labels during the training stage $\{l_0, l_1, \ldots, l_N\}$. Here a_i and r_i refer to the linear acceleration and angular rate measured by the IMU on the wrist at the i-th instance, respectively, while $l_i \in \{0, 1, \ldots, C\}$ represents the class ID for this specific frame. $C = 15$ is the number of classes to consider, and N refers to the frame number of one sequence. Note our current system supports IMU readings at frequencies up to 200 FPS. We choose to downsample the signals to rule out high-frequency details for better generalization ability. For every time instance t, we always combine the IMU readings of M previous frames and current one to form the input vector, namely $vec(a_{t-M}, a_{t-M+1}, \ldots, a_t, r_{t-M}, r_{t-M+1}, \ldots, r_t)$. Here *vec* means the vectorization of a list of numbers. Depending on the model used in the second stage, we may also apply some pre-processing on the input vector, like appending the frequency domain representation

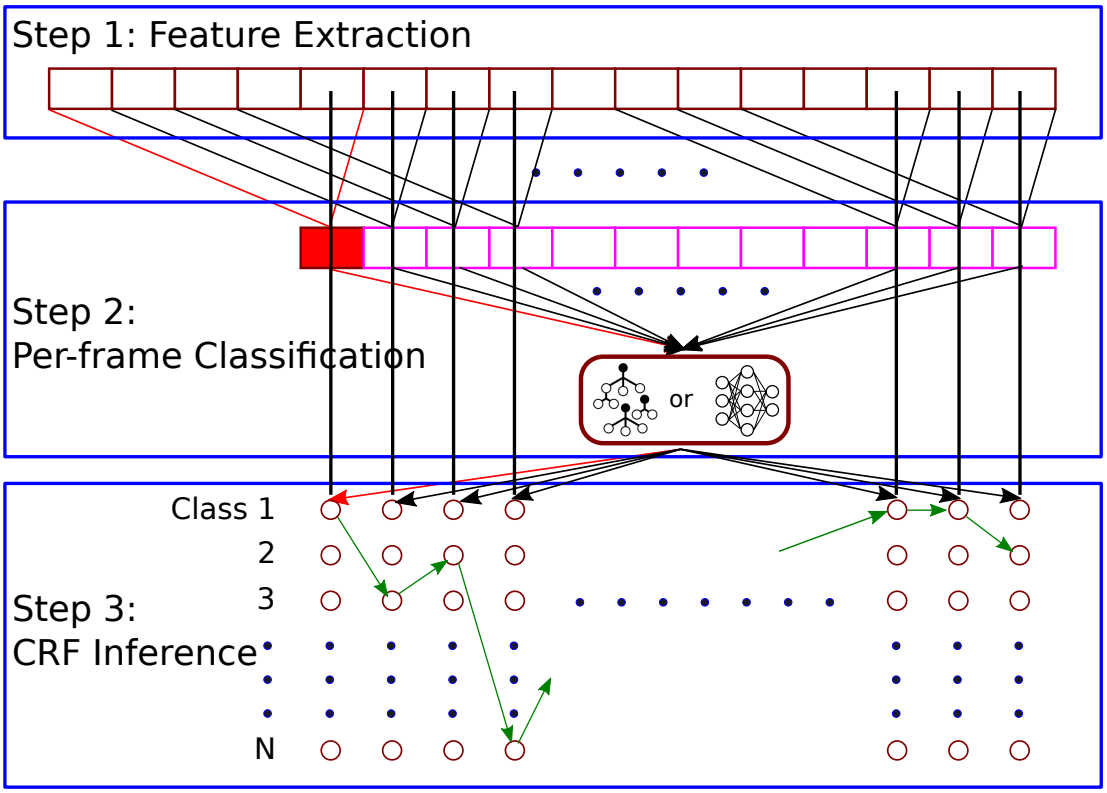


Figure 7.4: Our system is mainly composed of three components. In the feature extraction stage, IMU readings from continuous frames are concatenated to compose the input vector. Depending on the classification model used in the second stage, the raw signal and/or its frequency domain representation are fed. In this work, we mainly consider Random Forests and Multi-Layer-Perceptrons as the classifier. Given the class probability computed in the second stage, one extra linear-chain CRF inference is further introduced to encode temporal coherence in the last step.

of the measurement. In this case, the final input vector can be represented as:

$$cat(fft(vec(a_{t-M}, a_{t-M+1}, \ldots, a_t)), fft(vec(r_{t-M}, r_{t-M+1}, \ldots, r_t))) \quad (7.1)$$

where *fft* means Fast Fourier Transformation and *cat* is the vector concatenation operation. Since the aforementioned pre-processing is performed on all 3 channels of linear acceleration and angular rate, here for brevity we do not explicitly differentiate one specific axis of the measurements. The output from all 3 channels together make up the input vector to our system.

In the second stage, we conduct a per-frame classification of the features extracted as described. As the core of the whole system, this module plays a key role in determining the overall performance. We take a Neural Network as the underlying model due to its proven success in Computer Vision and Computer Graphics. As a baseline, we also tested a Random Forests. The experimental validation aligns with the general finding that Neural Network tends to behave better than Random Forests given sufficient data samples, while Random Forests already yields a quite strong result. More information about the network structure and training is given in the implementation section. After this step every single frame in the motion sequence is assigned a probability distribution $\{(p_0^0, p_1^0, \ldots, p_C^0), (p_0^1, p_1^1, \ldots, p_C^1), \ldots, (p_0^N, p_1^N, \ldots, p_C^N)\}$, where p_i^j is the estimated non-negative probability of j-th frame being classified as i-th class, and $\sum_{i=1}^{C} p_i^j = 1$ for any j.

By now, we have encoded the history information of the current frame by concatenating the IMU readings of its previous frames, which proves important to infer the attached semantic label. However, every frame is treated individually in the output class space, with no temporal coherence constraint applied. One possible consequence of this is a jittery prediction over the motion sequence. However, intuition suggests quite differently that natural human motion is usually smooth with reasonable rather than random transitions from one type of activity to another. For example, the activity Run is rarely preceded by the activity Eat. The similar idea has been widely adopted in Computer Vision to boost performance in topics like Image Segmentation (Zheng et al., 2015), Optical Flow Estimation (Sun et al., 2010), etc. Here we also try to utilize this straightforward yet general and empirically powerful prior information to regularize the initial output in the second stage. Refer to Figure 7.5 to make the concept clearer. Shown in the figure is the transition matrix we use through the experiments, which is computed from the training set captured in a continuous manner.

More specifically, the Viterbi decoding algorithm (Forney, 1973) proposed both in linear-chain Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) is used to refine the predicted per-frame results as one post-processing step. Given the probability distribution for every frame estimated in the second stage p_i^j, here we try to achieve a more temporally consistent solution that also eliminates non-natural transitions

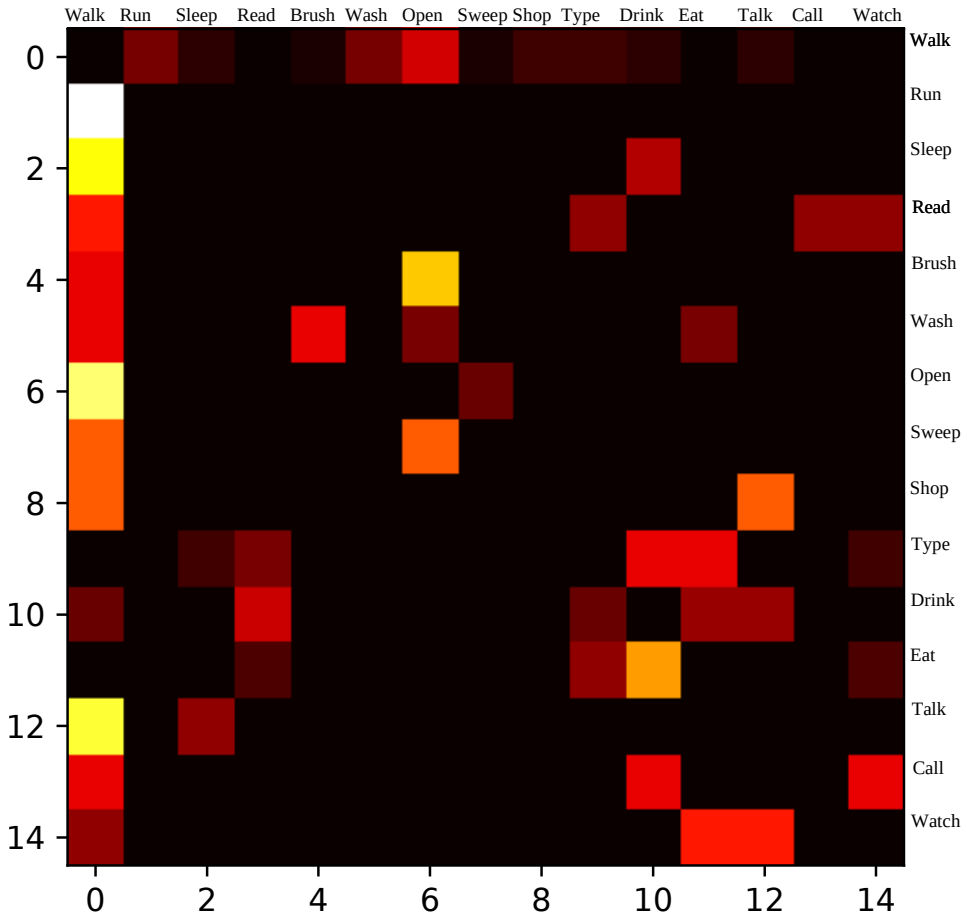


Figure 7.5: Visualization of the transition matrix computed from the continuous activity recording. This figure is color-coded: the warmer the color, the bigger the transition probability. Please note that the diagonal elements are set as zero to better manifest the transition pattern among different motions.

by optimizing the following objective function:

$$E(S) = \sum_{j=1}^{N} p_{S_j}^{j} + \lambda \sum_{j=1}^{N-1} T_{S_j,S_{j+1}} \tag{7.2}$$

where λ is the weighting parameter, T is the transition probability matrix computed from the training motion sequences captured in continuous mode, and $T_{i,j}$ indicates the likelihood of going from class i to class j over neighbouring frames. And $S = (S_0, S_1, \ldots, S_N)$ means the class labels assigned to the whole sequence, with $S_i \in (1, 2, \ldots, C)$. This operation is illustrated as step 3 in Figure 7.4.

7.4 Implementation Details

In this section, we specify the technical details involved in the implementation of our system. M is set as 60, namely, for every single frame, the IMU readings of 59 previous frames are combined with the current measurements to form the input vector. This makes the size of the raw input vector to be 360-d. The augmentation of the input vector by its *fft* representation is only used for the Random Forest since we find for the Neural Network, feeding the raw signal always yields comparable or even better classification results, manifesting the representation learning power of Neural Networks.

The whole system is implemented in Python[11]. For Random Forest, we use the implementation provided in Scipy (Bressert, 2012). We train and test the Neural Network structure via the publicly available PyTorch library (Paszke et al., 2017), one of the most popular auto-differential Deep Learning libraries. The structure of the Neural Network adopted in the experiments is shown in Figure 7.6. Other than the one showed here, we also try other versions that are deeper, wider, or with different activation functions and dropout rates. However, none of these boost performance in a significant way. For Viterbi decoding, we adopted the implementation in Tensorflow (Abadi et al., 2016b).

7.5 Experimental Results

We conduct extensive experiments to validate the proposed algorithm on the new DHIRI dataset. 6 of all 11 subjects are used for the model training, while the remaining subjects are held out for testing. We adopt the standard average accuracy as the metric to measure how well the model performs. Since the probability distribution of the prediction plays an important role in the temporal inference step, we also report the top 2 and top 3 average accuracies as an extra performance indicator of the per-frame classification. To evaluate the algorithm more completely, we also adjust different settings of the system, like only using one of linear acceleration and angular rate or both of them, and whether

[11] https://www.python.org

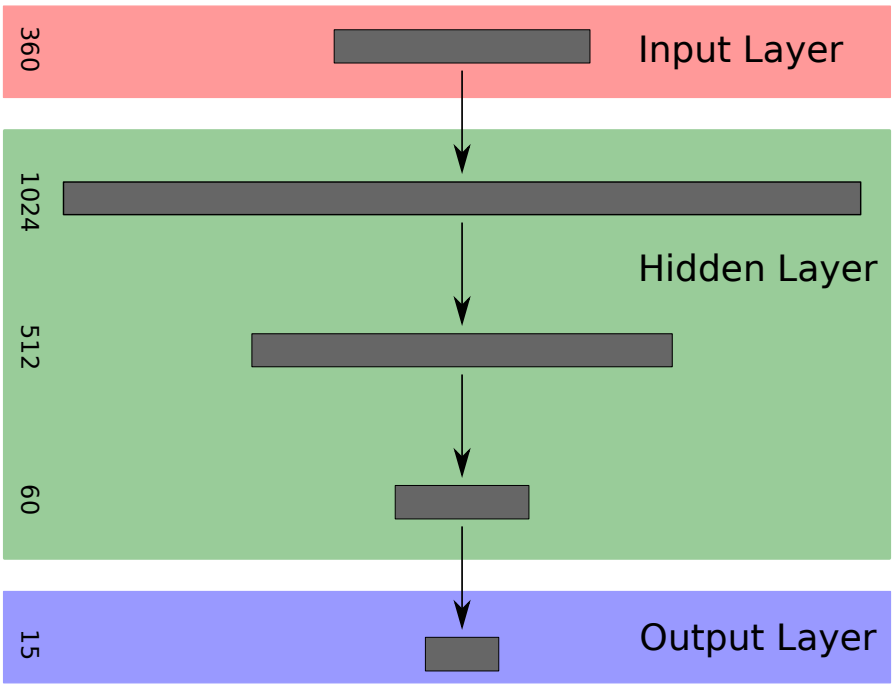


Figure 7.6: Network structure of the MLP used as the base classifier. There are three hidden layers sized 1024, 512 and 60, between the 360-d input and 15-d output layers. Each hidden layer is followed by a non-linear ReLU activation and Droput layer with 0.5 Dropout rate.

Setting	top-1	top-2	top-3	Final
Raw signal, MLP	0.61	0.74	0.81	0.74
Raw signal + FFT, MLP	0.61	0.74	0.81	0.74
Raw signal, RF	0.57	0.70	0.77	0.67
Raw signal + FFT, RF	0.58	0.71	0.80	0.69

Table 7.2: Classification results of our system on DRIHI dataset we construct. Here RS refers to only raw signal being used, MLP being Multi-Layer-Perceptron, RS+FFT meaning combination of raw signal and FFT output are used, RF donating Random Forests.

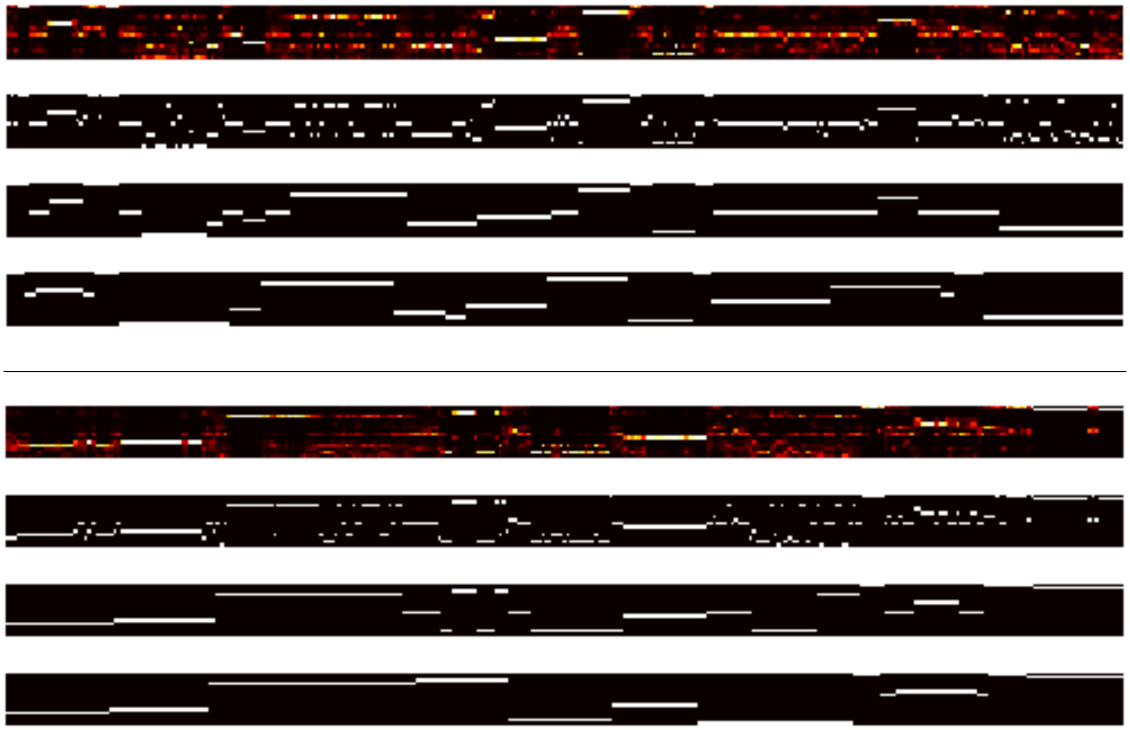


Figure 7.7: Illustration of the classification on two random testing sequences. For each column top-down: raw class probability distribution across the sequence obtained in the first stage; classification results of the first stage; results after linear-chain Conditional-Random-Fields (CRF) inference; the ground truth. Note how adding temporal coherence smooths and cleans the results by removing unnatural noise in the initial predictions.

to turn on *fft* pre-processing in the feature generation stage. The experimental results are shown in Table 7.2. Although Random Forest models are relatively easy to train and capable of obtaining decent results, an MLP yields better performance on our dataset. No matter which model is used, the combination of linear acceleration and angular rate steadily outperforms the versions using only one of them. It turns out that a strong correlation exists between top-1 accuracy and top-2/top-3 accuracy. The addition of temporal inference in the second stage always boosts the classification accuracy to a large extent (13% absolute performance gain). This phenomenon is more easily observed in the visualization of two random testing samples in Figure 7.7. We also show the confusion matrix before and after temporal inference for easier inspection of the result in Figure 7.8.

We also adjust different parts of the experimental settings to do ablation studies about our system. Since a better per-frame classification performance in the first stage consistently indicates a better final result, we only report the accuracy of the base classifier in the following experiments. We firstly train the MLP model on different subsets of the training data to see how the size of training samples influences the classification performance. The trained models are evaluated on the same held-out, thus unseen samples. The results are shown in Table 7.3. Not surprisingly, the classification accuracy is constantly boosted by adding more training data. It suggests that people do share some intrinsic way of performing the same activities, though there can be some inter-class variation.

We also consider the factor of the class number to predict. It is easy to imagine that the more classes to include and explain, the more difficult the problem is. To verify this

Train Size	top-1	top-2	top-3	Final
Ind_7 + Cond_4	0.61	0.74	0.81	0.74
Ind_7 + Cond_2	0.58	0.71	0.80	0.68
Ind_7	0.41	0.53	0.65	0.45
Ind_5	0.39	0.52	0.66	0.43
Ind_3	0.37	0.57	0.68	0.37
Ind_1	0.34	0.49	0.55	0.37

Table 7.3: Per-frame classification versus training size. Here Ind_D/CondD means the data captured with the first D subjects in individual (discrete) / continuous mode are used for training.

#Class	Top-1	Top-2	Top-3
15	0.61	0.74	0.81
12	0.72	0.82	0.88
10	0.78	0.88	0.92
8	0.88	0.95	0.97

Table 7.4: Number of class to predict versus per-frame classification accuracy. Note how reducing number of classes steadily improves the classification performance.

hypothesis, we adjust the number of activities, then only preserve the part of the data falling into the partially chosen motions. Here we keep the MLP network structure the same except for the last output layer. By changing the size of output nodes, we gain quite different classification performance, as shown in Table 7.4. We also vary the size of the temporal window to see how history information helps. For these experiments, we take different numbers of previous frames to combine with the current frame which makes the data sample, then re-train the same Neural Network by adjusting the size of input layer nodes accordingly. We find that a longer temporal window does benefit the base classifier before it saturates (see Table 7.5), and sampling frequency plays an important role. We hypothesize that an even larger dataset permits more complex and expressive models with larger temporal window size. We leave this for future work.

7.6 Conclusion and Future Work

In this work, we demonstrate the feasibility of human activity recognition using only one IMU-equipped smartwatch. To that end, we firstly capture one large-scale, purely realistic, and manually labeled dataset featuring 11 subjects conducting daily activities in everyday situations. Then we introduce a modulated data-driven model composed of per-frame classification and whole-sequence inference with temporal coherence and natural transition patterns considered. Our experimental settings are closer to reality than

#Setting	Top-1	Top-2	Top-3
10FPS, 6Sec	0.61	0.74	0.81
10FPS, 4Sec	0.61	0.74	0.81
10FPS, 2Sec	0.58	0.72	0.80
10FPS, 1Sec	0.55	0.68	0.77
5FPS, 6Sec	0.56	0.71	0.80
5FPS, 4Sec	0.56	0.71	0.80
5FPS, 2Sec	0.53	0.67	0.76
1FPS, 1Sec	0.34	0.48	0.59

Table 7.5: Accuracy versus sampling frequency and temporal length.

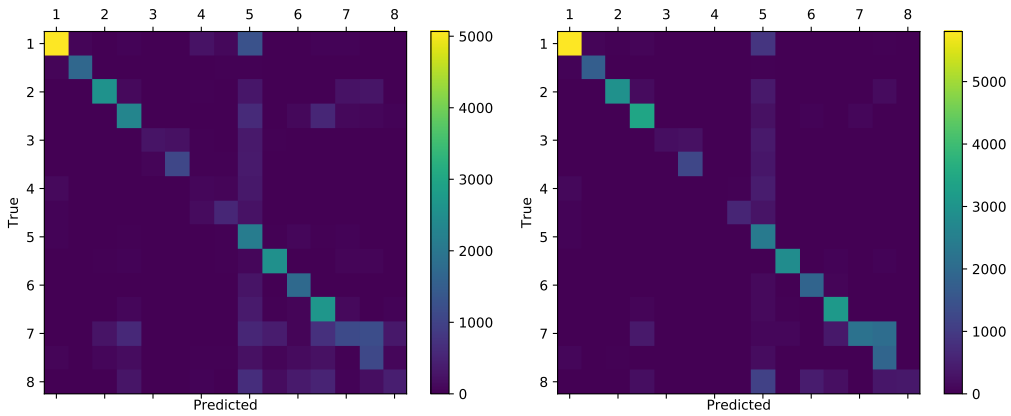


Figure 7.8: Confusion matrix before and after linear-chain CRF inference. Note the difference in the units in two matrices, and how adding temporal smoothness (right one) removes non-diagonal matrix elemnts.

previous work, and our proposed model better encodes natural constraints, thus giving better results. Our study also exposes the real threat of misuse and violation of personal privacy potentially introduced by widespread intelligent devices like smartwatches. One important thing to consider in future work is how to more efficiently record and label even larger motion data with no, or as few, external inferences as possible. A big enough dataset might admit other more sophisticated models like RNNs and LSTMs. It is also interesting to see how unsupervised learning or few-shot learning can be applied here. In our current study, we do not explicitly analyze the effect of cultural background on the way people behave, though the participants do span different nationalities. We believe extra effort should be spent in this direction.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In Computer Vision and Graphics, it is a long-standing goal to accurately capture natural human motion and interaction with objects inside diverse environments with little or even no intrusion. Targeting this goal in this thesis we contribute in three directions:

Fully Automatic Marker-less MoCap. In Chapter 4, we present the first fully automatic marker-less MoCap system that can naturally handle different body shapes and challenging poses. Unlike previous methods where the body model is manually constructed beforehand, our approach jointly optimizes the body shape and pose across the whole motion sequence. We demonstrate stable MoCap results comparable with, or even superior to, previous methods on standard benchmarks. The main driving force of our method is the state-of-the-art 3D generative body model and the DCT low-dimensional pose prior. Except for the 3D joint skeletons, our method also outputs a dense body surface mesh that is directly animatable.

Light-Weight MoCap at Interactive Rate. Compared with optimization-based MoCap methods that are usually quite time-consuming, learning-based alternatives promise the possibility of practical MoCap systems running at an interactive rate. Our method, Deep Inertial Poser, is an LSTM model that runs at 25FPS with adjustable time delay. Our method is lightweight in that it only requires 6 IMU sensors placed on different parts of the subject, thus is convenient to use. Learning-based methods are inherently data-hungry, while existing datasets with IMU readings and ground-truth body poses are limited in size and motion diversity. We propose a new method to synthesize IMU readings from an archival MoCap dataset. We show that it is effective to first train the model on the synthetic data, and then fine-tune it further on real data.

Joint Human and Object Tracking from RGB-D Sensors. Existing MoCap methods, especially the marker-less ones, mainly focus on the human body itself, while in reality, humans are frequently interacting with other objects. So it is naturally the next step to extend the body-only MoCap to jointly capture body motion and object motion, i.e., the interaction between humans and objects. We explore this direction by presenting the first system that can faithfully capture human-object interactions from 6 RGB-D cameras

arranged around the subject. We first treat the human and the object individually and fit them to 2D keypoints and contours obtained via CNN models. Then we jointly refine the human and object motions by introducing interpenetration and contact losses. Our key insight is that the object tends to stay in the same configuration relative to the hand during interaction.

8.2 Future Work

Though MoCap has achieved massive progress over the last decades, there is still a massive gap between what the current state-of-the-art permits and the requirements of modern VR/AR applications. The major problem to handle is still about improving the efficiency and effectiveness of MoCap with as little extra time and manual work as possible. Here we list some promising new trends:

MoCap from Neural Signals.

Until now, all the accurate MoCap methods rely on various body-mounted sensors to a different extent. To be applicable in daily life, the ideal MoCap systems should be even less intrusive. Among them, the neural signal reading sensors are the most interesting. Accurate hand pose estimation has been proved recently from CTRL-Labs Melcer et al. (2018), and it is naturally the next step to do a similar thing for the whole body.

MoCap with high-level Semantic Constraints.

As we all know, the human body pose is diverse and complex. This is one of the main reasons why MoCap is so difficult. However, across different subjects, the same kind of motions exhibit easily recognizable features. So it can be interesting to apply activity-specific motion priors when the subject is known to do some activities. Also, training a separate body pose/motion prior for each type of activities has a good chance of yielding better prior models than training a single model from all combined data.

MoCap among Multiple Subjects in Large Environments.

How to achieve accurate MoCap of multiple persons interacting with each other in large outdoor environments still remain a challenging problem, though a considerable degree of success has been achieved Guzov et al. (2021); Saini et al. (2019). Larger capture spaces and longer activities mean larger storage requirements and battery capability, which are difficult problems. Still, this direction is worth exploring given the increasing popularity of VR/AR, where many applications will need highly accurate, real-time, and lasting whole-body pose and/or appearance capture.

Bibliography

(2000). Cmu mocap dataset. http://mocap.cs.cmu.edu.

(2014). Chumpy. http://chumpy.org.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016a). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 308–318. ACM.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016b). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., and Li, H. (2019). Protecting world leaders against deep fakes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 38–45.

Aggarwal, J. K. and Ryoo, M. S. (2011). Human activity analysis: A review. ACM Computing Surveys (CSUR), **43**(3), 16.

Agre, P. E. and Rotenberg, M. (1998). Technology and privacy: The new landscape.

Ahmed, N., De Aguiar, E., Theobalt, C., Magnor, M., and Seidel, H.-P. (2005). Automatic generation of personalized human avatars from multi-view video. In Proceedings of the ACM symposium on Virtual reality software and technology, pages 257–260. ACM.

Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3d human pose reconstruction. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1446–1455.

Akhter, I., Sheikh, Y., Khan, S., and Kanade, T. (2010). Trajectory space: A dual representation for nonrigid structure from motion. IEEE Transactions on Pattern Analysis and Machine Intelligence, **33**(7), 1442–1456.

Akhter, I., Simon, T., Khan, S., Matthews, I., and Sheikh, Y. (2012). Bilinear spatiotemporal basis models. ACM Transactions on Graphics (TOG), **31**(2), 17.

Aksan, E., Kaufmann, M., and Hilliges, O. (2019). Structured prediction helps 3d human motion modelling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7144–7153.

Albright, R. K., Goska, B. J., Hagen, T. M., Chi, M. Y., Cauwenberghs, G., and Chiang, P. Y. (2011). Olam: A wearable, non-contact sensor for continuous heart-rate and activity monitoring. In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 5625–5628. IEEE.

Alldieck, T., Magnor, M., Xu, W., Theobalt, C., and Pons-Moll, G. (2018). Detailed human avatars from monocular video. In International Conference on 3D Vision (3DV).

Alldieck, T., Magnor, M., Bhatnagar, B. L., Theobalt, C., and Pons-Moll, G. (2019). Learning to reconstruct people in clothing from a single rgb camera. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1175–1186.

Amft, O., Junker, H., and Troster, G. (2005). Detection of eating and drinking arm gestures using inertial body-worn sensors. In Ninth IEEE International Symposium on Wearable Computers (ISWC'05), pages 160–163. IEEE.

Amin, S., Andriluka, M., Rohrbach, M., and Schiele, B. (2013). Multi-view pictorial structures for 3d human pose estimation. In BMVC. Citeseer.

Añazco, E. V., Lopez, P. R., Lee, S., Byun, K., and Kim, T.-S. (2018). Smoking activity recognition using a single wrist imu and deep learning light. In Proceedings of the 2nd international conference on digital signal processing, pages 48–51. ACM.

Andrews, S., Huerta, I., Komura, T., Sigal, L., and Mitchell, K. (2016). Real-time physics-based motion capture with sparse sensors. In Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016), page 5. ACM.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: shape completion and animation of people. In ACM SIGGRAPH 2005 Papers, pages 408–416.

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., and Amirat, Y. (2015). Physical human activity recognition using wearable sensors. Sensors, **15**(12), 31314–31338.

Bachlin, M., Roggen, D., Troster, G., Plotnik, M., Inbar, N., Meidan, I., Herman, T., Brozgol, M., Shaviv, E., Giladi, N., et al. (2009). Potentials of enhanced context awareness in wearable assistants for parkinson's disease patients with the freezing of gait syndrome. In 2009 International Symposium on Wearable Computers, pages 123–130. IEEE.

Bălan, A. O. and Black, M. J. (2008). The naked truth: Estimating body shape under clothing. In European Conference on Computer Vision, pages 15–29. Springer.

Balan, A. O., Sigal, L., Black, M. J., Davis, J. E., and Haussecker, H. W. (2007). Detailed human shape and pose from images. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE.

Baldominos, A., Saez, Y., and Isasi, P. (2018). Evolutionary design of convolutional neural networks for human activity recognition in sensor-rich environments. Sensors, **18**(4), 1288.

Ballan, L. and Cortelazzo, G. M. (2008). Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. 3DPVT, Atlanta, GA, USA, **37**.

Bao, L. and Intille, S. S. (2004). Activity recognition from user-annotated acceleration data. In International conference on pervasive computing, pages 1–17. Springer.

Barrat, J. (2013). Our final invention: Artificial intelligence and the end of the human era. Macmillan.

Barsoum, E., Kender, J., and Liu, Z. (2018). Hp-gan: Probabilistic 3d human motion prediction via gan. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 1418–1427.

Baumgart, B. G. (1974). Geometric modeling for computer vision. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Bayat, A., Pomplun, M., and Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. Procedia Computer Science, **34**, 450–457.

Bayati, H., Mill, J. d. R., Chavarriaga, R., et al. (2011). Unsupervised adaptation to on-body sensor displacement in acceleration-based activity recognition. In 2011 15th Annual International Symposium on Wearable Computers, pages 71–78. IEEE.

Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., and Ilic, S. (2014). 3d pictorial structures for multiple human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1669–1676.

Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. Spie.

Bhatnagar, B. L., Xie, X., Petrov, I. A., Sminchisescu, C., Theobalt, C., and Pons-Moll, G. (2022). BEHAVE: Dataset and method for tracking human object interactions.

Biggs, B., Ehrhadt, S., Joo, H., Graham, B., Vedaldi, A., and Novotny, D. (2020). 3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data. arXiv preprint arXiv:2011.00980.

Blanke, U. and Schiele, B. (2009). Daily routine recognition through activity spotting. In International Symposium on Location-and Context-Awareness, pages 192–206. Springer.

Bo, L. and Sminchisescu, C. (2010). Twin gaussian processes for structured prediction. International Journal of Computer Vision, **87**(1), 28–52.

Bogo, F., Romero, J., Loper, M., and Black, M. J. (2014). Faust: Dataset and evaluation for 3d mesh registration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3794–3801.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016a). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In Computer Vision – ECCV 2016, Lecture Notes in Computer Science, pages 561–578. Springer International Publishing.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016b). Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In European conference on computer vision, pages 561–578. Springer.

Bogo, F., Romero, J., Pons-Moll, G., and Black, M. J. (2017). Dynamic faust: Registering human bodies in motion. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6233–6242.

Bonomi, A. G., Goris, A. H., Yin, B., and Westerterp, K. R. (2009). Detection of type, duration, and intensity of physical activity using an accelerometer. Medicine & Science in Sports & Exercise, **41**(9), 1770–1777.

Boylan, J. F. (2018). Will deep-fake technology destroy democracy? The New York Times, Oct, **17**.

Bressert, E. (2012). SciPy and NumPy: an overview for developers. " O'Reilly Media, Inc.".

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). Numerical geometry of non-rigid shapes. Springer Science & Business Media.

Bulling, A., Blanke, U., and Schiele, B. (2014). A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys (CSUR), **46**(3), 33.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In CVPR.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2019). Openpose: realtime multi-person 2d pose estimation using part affinity fields. IEEE transactions on pattern analysis and machine intelligence, **43**(1), 172–186.

Cao, Z., Gao, H., Mangalam, K., Cai, Q.-Z., Vo, M., and Malik, J. (2020). Long-term human motion prediction with scene context. In European Conference on Computer Vision, pages 387–404. Springer.

Casas, L., Navab, N., and Demirci, S. (2019). Patient 3d body pose estimation from pressure imaging. International journal of computer assisted radiology and surgery, **14**(3), 517–524.

Chai, J. and Hodgins, J. K. (2005). Performance animation from low-dimensional control signals. In ACM Transactions on Graphics (TOG), volume 24, pages 686–696. ACM.

Chan, C., Ginosar, S., Zhou, T., and Efros, A. A. (2018). Everybody dance now. arXiv preprint arXiv:1808.07371.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012.

Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S. T., Tröster, G., Millán, J. d. R., and Roggen, D. (2013). The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. Pattern Recognition Letters, **34**(15), 2033–2042.

Chen, J., Kwong, K., Chang, D., Luk, J., and Bajcsy, R. (2006). Wearable sensors for reliable fall detection. In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, pages 3551–3554. IEEE.

Chen, K., Yao, L., Gu, T., Yu, Z., Wang, X., and Zhang, D. (2017a). Fullie and wiselie: A dual-stream recurrent convolutional attention model for activity recognition. arXiv preprint arXiv:1711.07661.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. (2016). Synthesizing training images for boosting human 3d pose estimation. In 2016 Fourth International Conference on 3D Vision (3DV), pages 479–488. IEEE.

Choutas, V., Pavlakos, G., Bolkart, T., Tzionas, D., and Black, M. J. (2020). Monocular expressive body regression through body-driven attention. arXiv preprint arXiv:2008.09062.

Corona, E., Pumarola, A., Alenya, G., Pons-Moll, G., and Moreno-Noguer, F. (2021). Smplicit: Topology-aware generative model for clothed people. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11875–11885.

De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. (2008). Performance capture from sparse multi-view video. In ACM Transactions on Graphics (TOG), volume 27, page 98. ACM.

De la Torre, F., Hodgins, J., Bargteil, A., Martin, X., Macey, J., Collado, A., and Beltran, P. (2008). Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Robotics Institute, page 135.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee.

Deutscher, J. and Reid, I. (2005). Articulated body motion capture by stochastic search. International Journal of Computer Vision, **61**(2), 185–205.

Deutscher, J., Blake, A., and Reid, I. (2000). Articulated body motion capture by annealed particle filtering. In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 2, pages 126–133. IEEE.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 2758–2766.

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., and Izadi, S. (2016). Fusion4d: Real-time performance capture of challenging scenes. ACM Trans. Graph., **35**(4), 114:1–114:13.

Du, Y., Wong, Y., Liu, Y., Han, F., Gui, Y., Wang, Z., Kankanhalli, M., and Geng, W. (2016). Marker-less 3d human motion capture with monocular image sequence and height-maps. In European Conference on Computer Vision, pages 20–36. Springer.

Elhayek, A., de Aguiar, E., Jain, A., Tompson, J., Pishchulin, L., Andriluka, M., Bregler, C., Schiele, B., and Theobalt, C. (2015). Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3810–3818. IEEE.

Fang, H.-S., Xie, S., Tai, Y.-W., and Lu, C. (2017). Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, pages 2334–2343.

Feng, Y., Choutas, V., Bolkart, T., Tzionas, D., and Black, M. J. (2021). Collaborative regression of expressive bodies using moderation. In 2021 International Conference on 3D Vision (3DV), pages 792–804. IEEE.

Ford, M. (2015). Rise of the Robots: Technology and the Threat of a Jobless Future. Basic Books.

Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, **61**(3), 268–278.

Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. (2015). Recurrent network models for human dynamics. In Computer Vision (ICCV), 2015 IEEE International Conference on, pages 4346–4354. IEEE.

Gall, J., Stoll, C., De Aguiar, E., Theobalt, C., Rosenhahn, B., and Seidel, H.-P. (2009). Motion capture using joint skeleton tracking and surface estimation. In Computer Vision and Pattern Recognition, 2009. CVPR 2009, pages 1746–1753.

Gall, J., Rosenhahn, B., Brox, T., and Seidel, H.-P. (2010). Optimization and filtering for human motion capture. International journal of computer vision, **87**(1-2), 75–92.

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 409–419.

Geman, S. (1987). Statistical methods for tomographic image reconstruction. Bull. Int. Stat. Inst, **4**, 5–21.

Geman, S. and McClure, D. E. (1987). Statistical methods for tomographic image reconstruction. In Proceedings of the 46th Session of the International Statistical Institute, Bulletin of the ISI, volume 52.

Geng, J., Shao, T., Zheng, Y., Weng, Y., and Zhou, K. (2018). Warp-guided gans for single-photo facial animation. In SIGGRAPH Asia 2018 Technical Papers, page 231. ACM.

Genova, K., Cole, F., Maschinot, A., Sarna, A., Vlasic, D., and Freeman, W. T. (2018). Unsupervised training for 3d morphable model regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8377–8386.

Ghosh, P., Song, J., Aksan, E., and Hilliges, O. (2017). Learning human motion models for long-term predictions. In 2017 International Conference on 3D Vision (3DV), pages 458–466. IEEE.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680.

Grzeszick, R., Lenk, J. M., Rueda, F. M., Fink, G. A., Feldhorst, S., and ten Hompel, M. (2017). Deep neural network based human activity recognition for the order picking process. In Proceedings of the 4th international Workshop on Sensor-based Activity Recognition and Interaction, page 14. ACM.

Guler, R. A. and Kokkinos, I. (2019). Holopose: Holistic 3d human reconstruction in-the-wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10884–10894.

Guzov, V., Mir, A., Sattler, T., and Pons-Moll, G. (2021). 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.

Habermann, M., Xu, W., Zollhoefer, M., Pons-Moll, G., and Theobalt, C. (2019). Livecap: Real-time human performance capture from monocular video. ACM Transactions On Graphics (TOG), **38**(2), 1–17.

Habermann, M., Xu, W., Zollhofer, M., Pons-Moll, G., and Theobalt, C. (2020). Deepcap: Monocular human performance capture using weak supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5052–5063.

Habermann, M., Xu, W., Zollhoefer, M., Pons-Moll, G., and Theobalt, C. (2021a). A deeper look into deepcap. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Habermann, M., Liu, L., Xu, W., Zollhoefer, M., Pons-Moll, G., and Theobalt, C. (2021b). Real-time deep dynamic characters. ACM Transactions on Graphics (TOG), **40**(4), 1–16.

Hammerla, N. Y., Halloran, S., and Plötz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv preprint arXiv:1604.08880.

Hampali, S., Rad, M., Oberweger, M., and Lepetit, V. (2020). Honnotate: A method for 3d annotation of hand and object poses. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3196–3206.

Hannink, J., Kautz, T., Pasluosta, C., Gassmann, K.-G., Klucken, J., and Eskofier, B. (2016). Sensor-based gait parameter extraction with deep convolutional neural networks. IEEE Journal of Biomedical and Health Informatics, pages 85–93.

Hasler, N., Ackermann, H., Rosenhahn, B., Thormählen, T., and Seidel, H.-P. (2010). Multilinear pose and body shape estimation of dressed subjects from image sets. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1823–1830. IEEE.

Hassan, M., Choutas, V., Tzionas, D., and Black, M. J. (2019). Resolving 3d human pose ambiguities with 3d scene constraints. In Proceedings of the IEEE International Conference on Computer Vision, pages 2282–2292.

Hassan, M., Ghosh, P., Tesch, J., Tzionas, D., and Black, M. J. (2021). Populating 3D scenes by learning human-scene interaction. pages 14708–14718.

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 11807–11816.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969.

Helten, T., Muller, M., Seidel, H.-P., and Theobalt, C. (2013). Real-time body tracking with one depth camera and inertial sensors. In Proceedings of the IEEE International Conference on Computer Vision, pages 1105–1112.

Hewitt, C., Bishop, P., and Steiger, R. (1973). A universal modular actor formalism for artificial intelligence. In Proceedings of the 3rd international joint conference on Artificial intelligence, pages 235–245. Morgan Kaufmann Publishers Inc.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal processing magazine, **29**.

Hirshberg, D. A., Loper, M., Rachlin, E., and Black, M. J. (2012). Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In European conference on computer vision, pages 242–255. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, **9**(8), 1735–1780.

Holden, D., Komura, T., and Saito, J. (2017). Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG), **36**(4), 42.

Huang, Y., Bogo, F., Lassner, C., Kanazawa, A., Gehler, P. V., Romero, J., Akhter, I., and Black, M. J. (2017). Towards accurate marker-less human shape and pose estimation over time. In 2017 international conference on 3D vision (3DV), pages 421–430. IEEE.

Huang, Y., Kaufmann, M., Aksan, E., Black, M. J., Hilliges, O., and Pons-Moll, G. (2018). Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. ACM Transactions on Graphics (TOG), **37**(6), 1–15.

Huang, Y., Taheri, O., Black, M. J., and Tzionas, D. (2022). Intercap: Joint markerless 3d tracking of humans and objects in interaction. In DAGM German Conference on Pattern Recognition, pages 281–299. Springer.

Ilic, S. and Fua, P. (2006). Implicit meshes for surface reconstruction. IEEE Transactions on Pattern Analysis and Machine Intelligence, **28**(2), 328–333.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2013). Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence, **36**(7), 1325–1339.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014a). Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence, **36**(7), 1325–1339.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014b). Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence, **36**(7), 1325–1339.

Jain, A., Thormählen, T., Seidel, H.-P., and Theobalt, C. (2010). Moviereshape: Tracking and reshaping of humans in videos. In ACM Transactions on Graphics (TOG), volume 29, page 148.

Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, pages 675–678.

Jiang, W., Kolotouros, N., Pavlakos, G., Zhou, X., and Daniilidis, K. (2020). Coherent reconstruction of multiple humans from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5579–5588.

Joo, H., Simon, T., and Sheikh, Y. (2018). Total capture: A 3d deformation model for tracking faces, hands, and bodies. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8320–8329.

Jordao, A., Nazare Jr, A. C., Sena, J., and Schwartz, W. R. (2018). Human activity recognition based on wearable sensor data: A standardization of the state-of-the-art. arXiv preprint arXiv:1806.05226.

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018a). End-to-end recovery of human shape and pose. In Computer Vision and Pattern Regognition (CVPR).

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018b). End-to-end recovery of human shape and pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7122–7131.

Karunratanakul, K., Yang, J., Zhang, Y., Black, M. J., Muandet, K., and Tang, S. (2020). Grasping field: Learning implicit representations for human grasps. In 2020 International Conference on 3D Vision (3DV), pages 333–344. IEEE.

Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3907–3916.

Kaufmann, M., Zhao, Y., Tang, C., Tao, L., Twigg, C., Song, J., Wang, R., and Hilliges, O. (2021). Em-pose: 3d human pose estimation from sparse electromagnetic trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11510–11520.

Kazemi, V., Burenius, M., Azizpour, H., and Sullivan, J. (2013). Multi-view body part recognition with random forests. In 2013 24th British Machine Vision Conference, BMVC 2013; Bristol; United Kingdom; 9 September 2013 through 13 September 2013. British Machine Vision Association.

Khedr, M. and El-Sheimy, N. (2017). A smartphone step counter using imu and magnetometer for navigation and health monitoring applications. Sensors, **17**(11), 2573.

Kim, H., Carrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., Pérez, P., Richardt, C., Zollhöfer, M., and Theobalt, C. (2018). Deep video portraits. ACM Transactions on Graphics (TOG), **37**(4), 163.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kirillov, A., Wu, Y., He, K., and Girshick, R. (2020). Pointrend: Image segmentation as rendering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9799–9808.

Kocabas, M., Athanasiou, N., and Black, M. J. (2020). 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.

Kolotouros, N., Pavlakos, G., Black, M. J., and Daniilidis, K. (2019). Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In Proceedings of the IEEE International Conference on Computer Vision.

Krishnan, A. (2016). Killer robots: legality and ethicality of autonomous weapons. Routledge.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

Langton, C. G. (1997). Artificial life: An overview. Mit Press.

Laput, G. and Harrison, C. (2019). Sensing fine-grained hand activity with smartwatches. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, page 338. ACM.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017a). Unite the people: Closing the loop between 3d and 2d human representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017b). Unite the people: Closing the loop between 3D and 2D human representations. In CVPR.

Learned-Miller, E., Huang, G. B., RoyChowdhury, A., Li, H., and Hua, G. (2016). Labeled faces in the wild: A survey. In Advances in face detection and facial image analysis, pages 189–248. Springer.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. nature, **521**(7553), 436.

Lester, J., Choudhury, T., and Borriello, G. (2006). A practical approach to recognizing physical activities. In International conference on pervasive computing, pages 1–16. Springer.

Li, H., Vouga, E., Gudym, A., Luo, L., Barron, J. T., and Gusev, G. (2013). 3d self-portraits. ACM Transactions on Graphics (TOG), **32**(6), 1–9.

Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.-S., and Lu, C. (2018). Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. arXiv preprint arXiv:1812.00324.

Li, T., Bolkart, T., Black, M. J., Li, H., and Romero, J. (2017). Learning a model of facial shape and expression from 4d scans. ACM Trans. Graph., **36**(6), 194–1.

Li, X., Liu, S., Kim, K., Wang, X., Yang, M.-H., and Kautz, J. (2019). Putting humans in a scene: Learning affordance in 3d indoor environments. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12368–12376.

Lin, K., Wang, L., and Liu, Z. (2021). End-to-end human pose and mesh reconstruction with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1954–1963.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer.

Liu, H., Wei, X., Chai, J., Ha, I., and Rhee, T. (2011). Realtime human motion control with a small number of inertial sensors. In Symposium on Interactive 3D Graphics and Games, pages 133–140. ACM.

Liu, S., Li, T., Chen, W., and Li, H. (2019). Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7708–7717.

Loper, M., Mahmood, N., and Black, M. J. (2014). Mosh: Motion and shape capture from sparse markers. ACM Trans. Graph., **33**(6), 220–1.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG), **34**(6), 1–16.

Loper, M. M. and Black, M. J. (2014). Opendr: An approximate differentiable renderer. In European Conference on Computer Vision, pages 154–169. Springer.

Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., and Black, M. J. (2020). Learning to dress 3d people in generative clothing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6469–6478.

Ma, Q., Saito, S., Yang, J., Tang, S., and Black, M. J. (2021). Scale: Modeling clothed humans with a surface codec of articulated local elements. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16082–16093.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). Amass: Archive of motion capture as surface shapes. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5442–5451.

Malleson, C., Volino, M., Gilbert, A., Trumble, M., Collomosse, J., and Hilton, A. (2017). Real-time full-body motion capture from video and imus. In 2017 Fifth International Conference on 3D Vision (3DV), pages 449–457.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.

Martinez, J., Black, M. J., and Romero, J. (2017). On human motion prediction using recurrent neural networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4674–4683. IEEE.

Maurer, U., Smailagic, A., Siewiorek, D. P., and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017a). Monocular 3d human pose estimation in the wild using improved cnn supervision. In 2017 international conference on 3D vision (3DV), pages 506–516. IEEE.

Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. (2017b). Vnect: Real-time 3d human pose estimation with a single rgb camera. volume 36.

Melcer, E. F., Astolfi, M. T., Remaley, M., Berenzweig, A., and Giurgica-Tiron, T. (2018). Ctrl-labs: Hand activity estimation and real-time control from neuromuscular signals. In Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pages 1–4.

Microsoft (2022). Azure Kinect SDK (K4A). https://github.com/microsoft/Azure-Kinect-Sensor-SDK.

Moreno-Noguer, F. (2016). 3d human pose estimation from a single image via distance matrix regression. arXiv preprint arXiv:1611.09010.

Mousas, C. (2017). Full-body locomotion reconstruction of virtual characters using a single inertial measurement unit. Sensors, **17**(11), 2589.

Moya Rueda, F., Grzeszick, R., Fink, G., Feldhorst, S., and ten Hompel, M. (2018). Convolutional neural networks for human activity recognition using body-worn sensors. In Informatics, volume 5, page 26. Multidisciplinary Digital Publishing Institute.

Nagano, K., Seo, J., Xing, J., Wei, L., Li, Z., Saito, S., Agarwal, A., Fursund, J., et al. (2018). pagan: real-time avatars using dynamic textures.

Nägeli, T., Oberholzer, S., Plüss, S., Alonso-Mora, J., and Hilliges, O. (2018). Flycon: real-time environment-independent multi-view human pose estimation with aerial vehicles. ACM Transactions on Graphics (TOG), **37**(6), 1–14.

Nocedal, J. and Wright, S. (2006a). Numerical optimization: Springer science & business media. New York.

Nocedal, J. and Wright, S. J. (2006b). Nonlinear equations. Numerical Optimization, pages 270–302.

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011). Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In 2011 International Conference on Computer Vision, pages 2088–2095. IEEE.

Ordóñez, F. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, **16**(1), 115.

Ordonez, F. J., Englebienne, G., De Toledo, P., Van Kasteren, T., Sanchis, A., and Kröse, B. (2014). In-home activity recognition: Bayesian inference for hidden markov models. IEEE Pervasive Computing, **13**(3), 67–75.

Osman, A. A., Bolkart, T., and Black, M. J. (2020). Star: Sparse trained articulated human body regressor. In European Conference on Computer Vision, pages 598–613. Springer.

Osman, A. A., Bolkart, T., Tzionas, D., and Black, M. J. (2022). Supr: A sparse unified part-based human representation. In European Conference on Computer Vision, pages 568–585. Springer.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pages 372–387. IEEE.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017a). Coarse-to-fine volumetric prediction for single-image 3D human pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017b). Harvesting multiple views for marker-less 3d human pose annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., and Black, M. J. (2019a). Expressive body capture: 3d hands, face, and body from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10975–10985.

Pavlakos, G., Kolotouros, N., and Daniilidis, K. (2019b). Texturepose: Supervising human mesh estimation with texture consistency. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 803–812.

Peng, X. B., Berseth, G., Yin, K., and Van De Panne, M. (2017). Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. ACM Trans. Graph., **36**(4), 41:1–41:13.

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. (2018). Sfv: Reinforcement learning of physical skills from videos. ACM Trans. Graph., **37**(6).

Peng, X. B., Ma, Z., Abbeel, P., Levine, S., and Kanazawa, A. (2021). Amp: Adversarial motion priors for stylized physics-based character control. ACM Trans. Graph., **40**(4).

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In CVPR'16.

Plankers, R. and Fua, P. (2003). Articulated soft objects for multi-view shape and motion capture. IEEE Transactions on Pattern Analysis and Machine Intelligence, **25**(CVLAB-ARTICLE-2003-003), 63–83.

Pons-Moll, G. (2014). Human Pose Estimation from Video and Inertial Sensors. Ph.D. thesis.

Pons-Moll, G., Baak, A., Helten, T., Müller, M., Seidel, H.-P., and Rosenhahn, B. (2010). Multisensor-fusion for 3d full-body human motion capture. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Pons-Moll, G., Baak, A., Gall, J., Leal-Taixe, L., Mueller, M., Seidel, H.-P., and Rosenhahn, B. (2011). Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In IEEE International Conference on Computer Vision (ICCV), pages 1243–1250.

Pons-Moll, G., Romero, J., Mahmood, N., and Black, M. J. (2015). Dyna: A model of dynamic human shape in motion. ACM Transactions on Graphics (TOG), **34**(4), 120.

Pons-Moll, G., Pujades, S., Hu, S., and Black, M. (2017). ClothCap: Seamless 4D clothing capture and retargeting. ACM Transactions on Graphics, **36**(4), 73:1–73:15.

Popa, A.-I., Zanfir, M., and Sminchisescu, C. (2017). Deep multitask architecture for integrated 2d and 3d human sensing. arXiv preprint arXiv:1701.08985.

Price, E., Lawless, G., Ludwig, R., Martinovic, I., Bülthoff, H. H., Black, M. J., and Ahmad, A. (2018). Deep neural network-based cooperative visual tracking through multiple micro aerial vehicles. IEEE Robotics and Automation Letters, **3**(4), 3193–3200.

Ramakrishna, V., Kanade, T., and Sheikh, Y. (2012). Reconstructing 3d human pose from 2d image landmarks. In European conference on computer vision, pages 573–586. Springer.

Ravi, D., Wong, C., Lo, B., and Yang, G.-Z. (2016). Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pages 71–76. IEEE.

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. (2020). Accelerating 3d deep learning with pytorch3d. arXiv preprint arXiv:2007.08501.

Reiss, A. and Stricker, D. (2012). Introducing a new benchmarked dataset for activity monitoring. In 2012 16th International Symposium on Wearable Computers, pages 108–109. IEEE.

Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., and Guibas, L. J. (2021). Humor: 3d human motion model for robust pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11488–11499.

Rhodin, H., Robertini, N., Richardt, C., Seidel, H.-P., and Theobalt, C. (2015). A versatile scene model with differentiable visibility applied to generative pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, pages 765–773.

Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.-P., Schiele, B., and Theobalt, C. (2016a). EgoCap: egocentric marker-less motion capture with two fisheye cameras. **35**(6), 162.

Rhodin, H., Robertini, N., Casas, D., Richardt, C., Seidel, H.-P., and Theobalt, C. (2016b). General automatic human shape and motion capture using volumetric contour cues. In European Conference on Computer Vision, pages 509–526. Springer.

Rippel, O., Snoek, J., and Adams, R. P. (2015). Spectral representations for convolutional neural networks. In Advances in neural information processing systems, pages 2449–2457.

Roetenberg, D., Luinge, H., and Slycke, P. (2007). Moven: Full 6dof human motion tracking using miniature inertial sensors. Xsen Technologies, December.

Roetenberg, D., Luinge, H., and Slycke, P. (2009). Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Xsens Motion Technologies BV, Tech. Rep, **1**.

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics (ToG), **36**(6), 1–17.

Ronao, C. A. and Cho, S.-B. (2015). Deep convolutional neural networks for human activity recognition with smartphone sensors. In International Conference on Neural Information Processing, pages 46–53. Springer.

Rowe, F. (2014). What literature review is not: diversity, boundaries and recommendations.

Saini, N., Price, E., Tallamraju, R., Enficiaud, R., Ludwig, R., Martinovic, I., Ahmad, A., and Black, M. J. (2019). Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 823–832.

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2304–2314.

Saito, S., Yang, J., Ma, Q., and Black, M. J. (2021). Scanimate: Weakly supervised learning of skinned clothed avatar networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2886–2897.

Savva, M., Chang, A. X., Hanrahan, P., Fisher, M., and Nießner, M. (2016). Pigraphs: learning interaction snapshots from observations. ACM Transactions on Graphics (TOG), **35**(4), 1–12.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, **45**(11), 2673–2681.

Schwartz, O. (2018). You thought fake news was bad? deep fakes are where truth goes to die. The Guardian.

Schwarz, L., Mateus, D., and Navab, N. (2009). Discriminative human full-body pose estimation from wearable inertial sensor data. Modelling the Physiological Human, pages 159–172.

Shimada, S., Golyanik, V., Xu, W., and Theobalt, C. (2020). Physcap: Physically plausible monocular 3d motion capture in real time. ACM Transactions on Graphics, **39**(6).

Shimada, S., Golyanik, V., Li, Z., Pérez, P., Xu, W., and Theobalt, C. (2022). Hulc: 3d human motion capture with pose manifold sampling and dense contact guidance. In European Conference on Computer Vision, pages 516–533. Springer.

Sigal, L., Balan, A., and Black, M. J. (2007). Combined discriminative and generative articulated pose and non-rigid shape estimation. In Advances in neural information processing systems, pages 1337–1344.

Sigal, L., Balan, A. O., and Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International journal of computer vision, **87**(1-2), 4.

Sigal, L., Isard, M., Haussecker, H., and Black, M. J. (2012). Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. International journal of computer vision, **98**(1), 15–48.

Simo-Serra, E., Quattoni, A., Torras, C., and Moreno-Noguer, F. (2013). A joint model for 2d and 3d pose estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3634–3641.

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 1145–1153.

Slyper, R. and Hodgins, J. (2008). Action capture with accelerometers. In Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '08, pages 193–199, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Starck, J. and Hilton, A. (2003). Model-based multiple view reconstruction of people. In null, page 915. IEEE.

Stoll, C., Hasler, N., Gall, J., Seidel, H.-P., and Theobalt, C. (2011). Fast articulated motion tracking using a sums of gaussians body model. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 951–958. IEEE.

Sun, D., Roth, S., and Black, M. J. (2010). Secrets of optical flow estimation and their principles. In 2010 IEEE computer society conference on computer vision and pattern recognition, pages 2432–2439. IEEE.

Sun, J., Wang, Z., Zhang, S., He, X., Zhao, H., Zhang, G., and Zhou, X. (2022). Onepose: One-shot object pose estimation without cad models. arXiv preprint arXiv:2205.12257.

Sun, Y., Liu, W., Bao, Q., Fu, Y., Mei, T., and Black, M. J. (2021). Putting people in their place: Monocular regression of 3d people in depth. arXiv preprint arXiv:2112.08274.

Sutton, R. S. and Barto, A. G. (2018). Reinforcement learning: An introduction.

Taheri, O., Ghorbani, N., Black, M. J., and Tzionas, D. (2020). Grab: A dataset of whole-body human grasping of objects. arXiv preprint arXiv:2008.11200.

Tao, Y., Zheng, Z., Guo, K., Zhao, J., Quionhai, D., Li, H., Pons-Moll, G., and Liu, Y. (2018). Doublefusion: Real-time capture of human performance with inner body shape from a depth sensor. In IEEE Conf. on Computer Vision and Pattern Recognition. CVPR Oral.

Tautges, J., Zinke, A., Krüger, B., Baumann, J., Weber, A., Helten, T., Müller, M., Seidel, H.-P., and Eberhardt, B. (2011). Motion reconstruction using sparse accelerometer data. ACM Transactions on Graphics (TOG), **30**(3), 18.

Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2015). Direct prediction of 3D body poses from motion compensated sequences. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., and Nießner, M. (2016). Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395.

Tiwari, G., Bhatnagar, B. L., Tung, T., and Pons-Moll, G. (2020). Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In European Conference on Computer Vision, pages 1–18. Springer.

Trumble, M., Gilbert, A., Hilton, A., and Collomosse, J. (2016). Deep convolutional networks for marker-less human pose estimation from multiple views. In Proceedings of CVMP 2016. The 13th European Conference on Visual Media Production.

Trumble, M., Gilbert, A., Malleson, C., Hilton, A., and Collomosse, J. P. (2017). Total capture: 3d human pose estimation fusing video and inertial sensors. In BMVC, volume 2, pages 1–13.

Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., and Gall, J. (2016). Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision, **118**(2), 172–193.

Valarezo, E., Rivera, P., Park, J., Gi, G., Kim, T., Al-Antari, M., Al-Masni, M., and Kim, T. (2017). Human activity recognition using a single wrist imu sensor via deep learning convolutional and recurrent neural nets. UNIKOM Journal of ICT, Design, Engineering and Technological Science1, **1**, 1–5.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. CoRR, **abs/1609.03499**.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 109–117.

Vertens, J., Fischer, F., Heyde, C., Hoeflinger, F., Zhang, R., Reindl, L. M., and Gollhofer, A. (2015). Measuring respiration and heart rate using two acceleration sensors on a fully embedded platform.

Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., and Popović, J. (2007). Practical motion capture in everyday surroundings. volume 26, page 35. ACM.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. In ACM Transactions on Graphics (TOG), volume 27, page 97. ACM.

Von Marcard, T., Pons-Moll, G., and Rosenhahn, B. (2016). Human pose estimation from video and imus. IEEE transactions on pattern analysis and machine intelligence, **38**(8), 1533–1547.

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

Von Marcard, T., Rosenhahn, B., Black, M. J., and Pons-Moll, G. (2017). Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In Computer Graphics Forum, volume 36, pages 349–360. Wiley Online Library.

von Marcard, T., Henschel, R., Black, M., Rosenhahn, B., and Pons-Moll, G. (2018a). Recovering accurate 3d human pose in the wild using imus and a moving camera. In European Conference on Computer Vision (ECCV).

von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., and Pons-Moll, G. (2018b). Recovering accurate 3d human pose in the wild using imus and a moving camera. In Proceedings of the European Conference on Computer Vision (ECCV), pages 601–617.

Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L. (2017). Deep learning for sensor-based activity recognition: A survey. arXiv preprint arXiv:1707.03502.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4724–4732.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. The knowledge engineering review, **10**(2), 115–152.

Wu, C., Varanasi, K., and Theobalt, C. (2012). Full body performance capture under uncontrolled and varying illumination: A shading-based approach. In European Conference on Computer Vision, pages 757–770. Springer.

Xiang, S. (2021). Eliminating topological errors in neural network rotation estimation using self-selecting ensembles. ACM Transactions on Graphics (TOG), **40**(4), 1–21.

Xiu, Y., Li, J., Wang, H., Fang, Y., and Lu, C. (2018). Pose Flow: Efficient online pose tracking. In BMVC.

Xu, H., Bazavan, E. G., Zanfir, A., Freeman, W. T., Sukthankar, R., and Sminchisescu, C. (2020). Ghum & ghuml: Generative 3d human shape and articulated pose models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6184–6193.

Xu, W., Avishek, C., Zollhöfer, M., Rhodin, H., Mehta, D., Seidel, H.-P., and Theobalt, C. (2017). Monoperfcap: Human performance capture from monocular video. arXiv preprint arXiv:1708.02136.

Xu, W., Chatterjee, A., Zollhöfer, M., Rhodin, H., Mehta, D., Seidel, H.-P., and Theobalt, C. (2018a). Monoperfcap: Human performance capture from monocular video. ACM Transactions on Graphics (ToG), **37**(2), 1–15.

Xu, W., Chatterjee, A., Zollhoefer, M., Rhodin, H., Mehta, D., Seidel, H.-P., and Theobalt, C. (2018b). Monoperfcap: Human performance capture from monocular video.

Yang, D., Huang, J., Tu, X., Ding, G., Shen, T., and Xiao, X. (2018). A wearable activity recognition device using air-pressure and imu sensors. IEEE Access, **7**, 6611–6621.

Yang, J., Nguyen, M. N., San, P. P., Li, X. L., and Krishnaswamy, S. (2015). Deep convolutional neural networks on multichannel time series for human activity recognition. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

Yao, A., Gall, J., Gool, L. V., and Urtasun, R. (2011). Learning probabilistic nonlinear latent variable models for tracking complex activities. In Advances in Neural Information Processing Systems, pages 1359–1367.

Yao, B. and Fei-Fei, L. (2010). Modeling mutual context of object and human pose in human-object interaction activities. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 17–24. IEEE.

Yao, R., Lin, G., Shi, Q., and Ranasinghe, D. C. (2018). Efficient dense labelling of human activity sequences from wearables using fully convolutional networks. Pattern Recognition, **78**, 252–266.

Yasin, H., Iqbal, U., Kruger, B., Weber, A., and Gall, J. (2016). A dual-source approach for 3d pose estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4948–4956.

Yi, H., Huang, C.-H. P., Tzionas, D., Kocabas, M., Hassan, M., Tang, S., Thies, J., and Black, M. J. (2022). Human-aware object placement for visual environment reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3959–3970.

Zanfir, A., Bazavan, E. G., Xu, H., Freeman, W. T., Sukthankar, R., and Sminchisescu, C. (2020). Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In European Conference on Computer Vision, pages 465–481. Springer.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J. (2014). Convolutional neural networks for human activity recognition using mobile sensors. In 6th International Conference on Mobile Computing, Applications and Services, pages 197–205. IEEE.

Zhang, C., Pujades, S., Black, M. J., and Pons-Moll, G. (2017). Detailed, accurate, human shape estimation from clothed 3d scan sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4191–4200.

Zhang, J. Y., Pepose, S., Joo, H., Ramanan, D., Malik, J., and Kanazawa, A. (2020a). Perceiving 3d human-object spatial arrangements from a single image in the wild. In European Conference on Computer Vision (ECCV).

Zhang, S., Zhang, Y., Bogo, F., Pollefeys, M., and Tang, S. (2021a). 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.

Zhang, Y., An, L., Yu, T., Li, X., Li, K., and Liu, Y. (2020b). 4d association graph for realtime multi-person motion capture using multiple video cameras. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1324–1333.

Zhang, Y., Hassan, M., Neumann, H., Black, M. J., and Tang, S. (2020c). Generating 3d people in scenes without people. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6194–6204.

Zhang, Y., Li, Z., An, L., Li, M., Yu, T., and Liu, Y. (2021b). Lightweight multi-person total motion capture using sparse multi-view cameras. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5560–5569.

Zhao, M., Liu, Y., Raghu, A., Li, T., Zhao, H., Torralba, A., and Katabi, D. (2019). Through-wall human mesh recovery using radio signals. In Proceedings of the IEEE International Conference on Computer Vision, pages 10113–10122.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In Proceedings of the IEEE international conference on computer vision, pages 1529–1537.

Zheng, Y., Shao, R., Zhang, Y., Yu, T., Zheng, Z., Dai, Q., and Liu, Y. (2021). Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6239–6249.

Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016). Sparseness meets deepness: 3d human pose estimation from monocular video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4966–4975.

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. (2019). On the continuity of rotation representations in neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5745–5753.

Zhou, Y., Habermann, M., Habibie, I., Tewari, A., Theobalt, C., and Xu, F. (2021). Monocular real-time full body capture with inter-part correlations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4811–4822.

Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., and Kolb, A. (2018). State of the art on 3d reconstruction with rgb-d cameras. In Computer graphics forum, volume 37, pages 625–652. Wiley Online Library.