PHYSICALLY PLAUSIBLE 3D HUMAN MOTION CAPTURE AND SYNTHESIS WITH INTERACTIONS

SOSHI SHIMADA

Dissertation zur Erlangung des Grades des

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

Saarbrücken, 2024

Date of Colloquium:	April 4, 2024
Dean of the Faculty:	Prof. Dr. Roland Speicher
Chair of the Committee:	Prof. Dr. Philipp Slusallek
Reviewers:	Prof. Dr. Christian Theobalt
	Dr. Patrick Pérez
	Prof. Dr. Taku Komura
Academic Assistant:	Dr. Rishabh Dabral

ABSTRACT

Modelling 3D human motion is highly important in numerous applications, including AR/VR, human-robot interaction, gaming, and character animations. To develop such applications, plausible 3D human motions need to be captured from sensing devices or synthesised based on the motion model definition.

Obtaining 3D human motion from a single RGB camera is one of the ideal setups for motion capture due to its flexibility in the recording locations and the subject's clothes, and cost-effectiveness, unlike heavy setups such as marker-based or marker-less multi-view motion capture systems. However, capturing the 3D motions only from a monocular camera is a highly ill-posed problem, which can result in the implausible reconstruction of the motions (*e.g.* jitter, foot-skating, unnatural body leaning and inaccurate 3D localisations). The problem becomes more challenging when considering interactions with environments and surface deformations; The human body's occlusions and the lack of modelling for the interactions and deformations often lead to physically implausible collisions. Therefore, the captured motions often require costly and time-consuming manual post-processing by experts before integration into industry products.

Another major approach for obtaining 3D human motions is through the use of motion synthesis methods. While many learning-based 3D motion synthesis works have been proposed — including those that can consider hand-hand and/or hand-object interactions — they often lack realism. Many synthesis methods consider the shape and semantics of the interacting object/environment. However, one crucial aspect missing from current methods is the consideration of physical quantities. For example, in our daily lives, our behaviour can be significantly influenced by the physical properties of objects, such as their mass. No prior works have explicitly addressed this factor when synthesising 3D motions.

This thesis addresses the aforementioned problems for motion capture with a monocular RGB camera and motion synthesis considering a physics quantity. First, a monocular video-based MoCap method with the explicit integration of rigid body dynamics modelling is proposed,

mitigating the artefacts typically observable in the existing kinematics-based MoCap methods. To introduce the power of learned physics prior, the fully learning physics-based MoCap method is proposed next. It highly improves the 3D accuracies while suppressing the artefacts in the reconstructed motions thanks to the network components trained with explicit physics modelling. Third, MoCap with interactions in a complex scene such as indoors with occluding objects is addressed. By modelling the whole-body contact with the environment and introducing a novel collision handling component, the plausibility of interactions in the captured motion is greatly improved compared with the prior works. Moreover, this thesis presents the first method that captures not only the hand and face motions but also the deformations arising from their interactions, which is of high importance for various Graphics applications that require immersive experiences. Furthermore, a novel 3D motion synthesis method is proposed next. This method generates 3D object manipulations with hands that exhibit realistic motions and interactions, plausibly adapting to the conditioning object's mass. Additionally, the method can optionally take a user-provided object trajectory as input and synthesise natural object manipulations influenced by the object's mass, offering a potential for substantial contributions to computer graphics applications. Lastly, the insights collected in this thesis and the outlook of the human motion capture and synthesis research are discussed.

The introduced methods in this thesis serve as milestones toward democratising the realistic low-cost human motion capture that replaces the aforementioned heavy motion capture setups and toward the widespread use of learning-based motion synthesis methods in industrial applications that require high motion realism.

ZUSAMMENFASSUNG

Die Modellierung menschlicher 3D-Bewegungen ist für zahlreiche Anwendungen von großer Bedeutung, darunter AR/VR, Mensch-Roboter-Interaktion, Spiele und Charakteranimationen. Um solche Anwendungen zu entwickeln, müssen plausible menschliche 3D-Bewegungen von Erfassungsgeräten erfasst oder auf der Grundlage der Definition des Bewegungsmodells synthetisiert werden.

Die Erfassung menschlicher 3D-Bewegungen mit einer einzigen RGB-Kamera ist eines der idealen Setups für die Bewegungserfassung, da es flexibel in Bezug auf die Aufnahmeorte und die Kleidung des Probanden ist und kostengünstig, im Gegensatz zu schweren Setups wie markerbasierten oder markerlosen Multi-View-Bewegungserfassungssystemen. Die Erfassung der 3D-Bewegungen nur mit einer monokularen Kamera ist jedoch ein äußerst ungünstiges Problem, das zu einer unplausiblen Rekonstruktion der Bewegungen führen kann (Zittern, Fußbewegungen, unnatürliche Körperneigung und ungenaue 3D-Lokalisierung). Das Problem wird noch schwieriger, wenn Interaktionen mit der Umgebung und Oberflächenverformungen berücksichtigt werden. Die Verdeckung des menschlichen Körpers und die fehlende Modellierung der Interaktionen und Verformungen führen oft zu physikalisch unplausiblen Kollisionen. Aus diesem Grund müssen die erfassten Bewegungen vor der Integration in Industrieprodukte oft kosten- und zeitaufwändig manuell von Experten nachbearbeitet werden.

Ein weiterer wichtiger Ansatz zur Gewinnung menschlicher 3D Bewegungen ist die Verwendung von Bewegungssynthesemethoden. Obwohl viele lernbasierte 3D-Bewegungssynthesemethoden vorgeschlagen wurden - einschließlich solcher, die Hand-Hand- und/oder Hand-Objekt-Interaktionen berücksichtigen können - mangelt es ihnen oft an Realismus. Viele Synthesemethoden berücksichtigen die Form und die Semantik des interagierenden Objekts/Umfelds. Ein entscheidender Aspekt, der bei den derzeitigen Methoden fehlt, ist jedoch die Berücksichtigung physikalischer Größen. In unserem täglichen Leben kann unser Verhalten beispielsweise erheblich von den physikalischen Eigenschaften von

Objekten, wie ihrer Masse, beeinflusst werden. Bisherige Arbeiten haben diesen Faktor bei der Synthese von 3D-Bewegungen nicht explizit berücksichtigt.

Diese Arbeit befasst sich mit den oben genannten Problemen bei der Bewegungserfassung mit einer monokularen RGB-Kamera und der Bewegungssynthese unter Berücksichtigung eines physikalischen Bewegungsmodells.

Zunächst wird eine monokulare videobasierte MoCap-Methode mit der expliziten Integration der Starrkörperdynamikmodellierung vorgeschlagen, die die Artefakte, die typischerweise bei den bestehenden kinematikbasierten MoCap-Methoden zu beobachten sind, abmildert. Um die Leistungsfähigkeit der erlernten Physikpriorität einzuführen, wird als nächstes die vollständig lernende, physikbasierte MoCap-Methode vorgeschlagen. Sie verbessert die 3D-Genauigkeit erheblich und unterdrückt gleichzeitig die Artefakte in den rekonstruierten Bewegungen dank der mit expliziter Physikmodellierung trainierten Netzwerkkomponenten. Drittens wird MoCap mit Interaktionen in einer komplexen Szene, z. B. in Innenräumen mit verdeckten Objekten, behandelt. Durch die Modellierung des Kontakts des Koerpers mit der Umgebung und die Einführung einer neuartigen Komponente zur Kollisionsbehandlung wird die Plausibilität der Interaktionen in der erfassten Bewegung im Vergleich zu früheren Arbeiten erheblich verbessert. Darüber hinaus wird in dieser Arbeit die erste Methode vorgestellt, die nicht nur die Hand- und Gesichtsbewegungen, sondern auch die aus ihren Interaktionen resultierenden Verformungen erfasst, was für verschiedene Grafikanwendungen, die immersive Erfahrungen erfordern, von großer Bedeutung ist. Außerdem wird eine neuartige 3D-Bewegungssynthesemethode vorgeschlagen. Diese Methode erzeugt 3D-Objektmanipulationen mit Händen, die realistische Bewegungen und Interaktionen aufweisen und sich plausibel an die Masse des konditionierten Objekts anpassen. Darüber hinaus kann die Methode optional eine vom Benutzer bereitgestellte Objekttrajektorie als Eingabe verwenden und natürliche Objektmanipulationen synthetisieren, die von der Masse des Objekts beeinflusst werden, was einen wesentlichen Beitrag zu Computergrafikanwendungen leisten könnte. Abschließend werden die in dieser Arbeit gewonnenen Erkenntnisse und der Ausblick eventuell weitere offene Forschungsfragen im Bereich der menschlichen Bewegungserfassung und -synthese diskutiert.

Die in dieser Arbeit vorgestellten Methoden dienen als Meilensteine auf dem Weg zur Demokratisierung der realistischen und kostengünstigen Erfassung menschlicher Bewegungen, die die oben erwähnten schwerfälligen Bewegungserfassungssysteme ersetzen, und auf dem Weg zum weit verbreiteten Einsatz von lernbasierten Bewegungssynthesemethoden in industriellen Anwendungen, die einen hohen Bewegungsrealismus erfordern.

ACKNOWLEDGMENTS

I wish to extend my heartfelt gratitude to all who have supported me throughout my PhD journey.

First and foremost, I would like to express my immense gratitude to my PhD adviser, Christian Theobalt, for offering me an amazing opportunity to start my research and for supporting me in nurturing my research skills. I learned a lot from Christian as he is not only a great researcher and mentor but also an inspiring leader who manages quite a large group of researchers. I am also deeply appreciative of the amazing research facility he offered, from which I could do a lot of cutting-edge research. I cannot thank Vladislav Golyanik enough for his dedicated support and supervision, starting from my master's project through to my PhD research. Our shared experiences outside of work have enriched my journey, making it all the more memorable. I also wish to acknowledge Weipeng Xu and Patrick Pérez. Their invaluable insights and active collaborations were essential for the success of my research projects. Their contributions made presenting at top-tier conferences a reality. I extend my gratitude to administrative members Ellen Fries, Sabine Budde, Alina Ashraf, tech admin Gereon Fox, Pramod Ramesh Rao, and IST members for always kindly taking care of my requests. Their assistance ensured the smooth execution of logistics and research activities. Sincere thanks go to Ikhsanul Habibie, Rishabh Dabral, Mallikarjun B R, Navami Kairanda, Wanyue Zhang and Mohit Mendiratta. Their companionship during both the joyous and challenging times has been a guiding light. The moments we have shared outside the work have added layers to my personal growth. To Jigyasa Katrolia, Ahmed Elsherif, Nowras Altaleb, and Shama Bhosale, thank you for always caring about my life, despite the distance that separates us. Your reminders to balance work with leisure often provided the respite I needed. Special thanks go to Franziska Mueller, Jan Bednařík and Thabo Beeler for hosting my internship at Google Zurich. Their efforts ensured a seamless transition and a rewarding research experience in a new city. To all the lab members, whether it was a casual chat or a heated academic discussion, each interaction with you

all has been enlightening. Your collective wisdom and camaraderie have enriched my PhD journey beyond words. Lastly, my deepest gratitude goes to my family. Your unwavering love and support have been the foundation upon which all my achievements stand. Without you, none of these would have been possible.

Thank you all for being a significant part of this chapter in my life.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

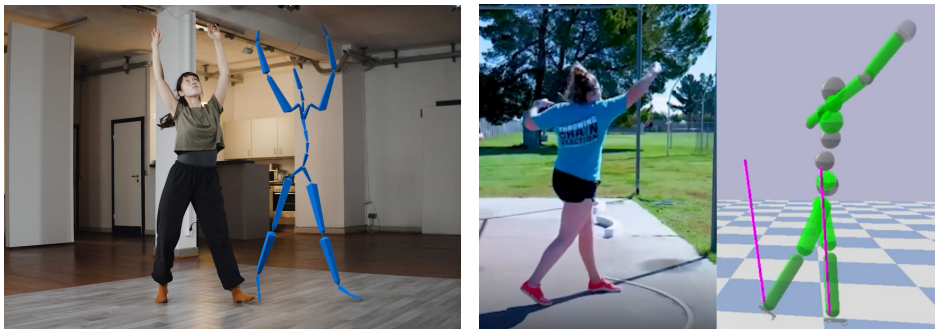
1 INTRODUCTION

1.1 MOTIVATION

3D human motion modelling is a mathematical representation of human movements in 3D space, typically modelled as a sequence of 3D joint angles of a kinematic skeleton, and is highly important for various computer graphics and vision applications. To utilise the 3D motions for real-world applications, they need to be either crafted manually, captured from sensors or synthesised based on the modelling definition.

In industry and academic settings, human motions are often obtained using a motion capture system (MoCap). MoCap is an active area of study focused on obtaining human movements, typically achieved using devices such as RGB cameras, depth sensors, or systems with optical markers attached to the subject's body. Captured realistic 3D human motions are invaluable in driving animated characters, significantly reducing the cost and production time in video games or film production. Additionally, these captured motions find applications in AR and VR environments, enabling the virtual presence of real persons in them. They also serve a critical role in sports analysis, helping athletes and trainers refine skills.

The prevalent method for MoCap in the industry involves subjects wearing suits equipped with optical markers. While this approach provides highly accurate motion capture, it is inflexible in recording locations and expensive. Furthermore, these systems do not permit subjects to wear everyday clothing during the capture, presenting significant drawbacks, especially when developing a learning-based system aimed at generalising across everyday life scenes that accepts visual inputs such as 2D images. In response to these limitations, markerless multi-view RGB camera-based motion capture systems are gaining popularity (see Fig. 1.1a). These systems enable the capture of highly accurate 3D motions of subjects in their everyday clothing. Nonetheless, both marker-based and multi-view RGB camera systems are resource-intensive, costly and impose limitations on recording locations. Consequently, there is a natu-



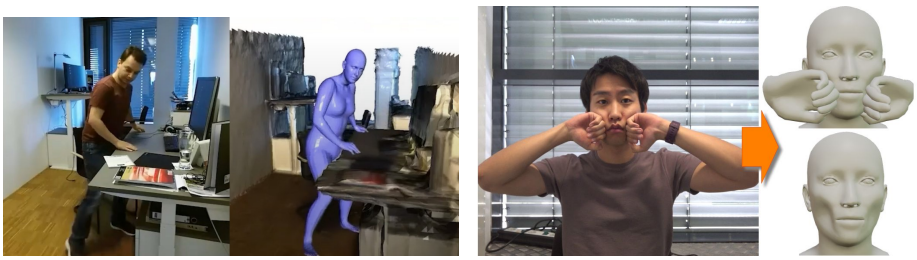
(a) Markerless MoCap (*The Captury* 2023)

(b) Torque Visualisation in MoCap (Shimada et al., 2021)

Figure 1.1: (a) Multi-view RGB based 3D motion capture system that reconstructs skeletal motions which are used in various applications such as AR/VR, filmmaking, etc (*The Captury* 2023). (b) Visualisation of torques (green) and the ground reaction force (purple) obtained from the physics-based markerless human motion capture system (Shimada et al., 2021), which can provide a deeper insight of human motions combined with the reconstructed skeletal motions.

ral motivation to explore methods that reduce setup requirements such as RGB cameras, IMU sensors, depth sensors, or combinations thereof.

Among the MoCap setups, capturing 3D human motions only from a single RGB camera is one of the least constraining setups for MoCap, thus it has been gaining a lot of attention in computer graphics and vision research communities. However, this simplification of the capture setup introduces significant challenges for accurately reconstructing 3D human motions. Given only a single view, it is theoretically impossible to simultaneously find the correct scale and depth of the object in the frame. Furthermore, self- and external- occlusions are even more prominent in a single-view setup compared to a multi-view scenario, *e.g.* only one side of the body is visible from a single camera. These difficulties result in artefacts in the reconstructed 3D motions, such as joint jitter, unnaturally leaning bodies, irregular joint angles, self-collisions, environmental collisions, and foot skating. Although the flexibility of the capture location is an advantage of the lightweight monocular setup, capturing the 3D motions becomes even more challenging when the motions are performed in complex environments, such as indoor settings with many occluding objects (see Fig. 1.2a). Such severe occlusions can lead to inaccuracy in the captured motions. Moreover, when the 3D scene geometry is given, reconstructing highly intricate interactions with the environment is re-



(a) MoCap in an indoor scene (Shimada et al., 2022)

(b) MoCap with deformations (Shimada et al., 2023)

Figure 1.2: Examples of new MoCap methods with external-/self-interactions. (a) the method that considers everyday interactions in an indoor scene (Shimada et al., 2022). (b) the method that takes into account self-interactions and their deformations of a face surface, which are frequently seen in our daily lives (Shimada et al., 2023).

quired; even a 3D localisation error of 1 cm can lead to highly implausible body-environment collisions, particularly around interaction regions with the environment. Furthermore, capturing only 3D joint angles overlooks many aspects. Real-world interactions often involve deformations of our body surfaces due to the soft tissues of the human body (see Fig. 1.2b). Such aspects also need to be captured for improved realism, vital for many graphics applications such as computer-animated characters and avatars for telepresence.

As an alternative technique to obtain 3D human motion, learning-based motion synthesis research is also coming increasingly into focus. Numerous methods have been developed that generate 3D motions controlled by text descriptions, action labels, or auditory signals. Recent advancements even account for interactions with objects or surrounding environments. Yet, a significant oversight is that none of these techniques fully integrate the object's inherent physical properties, which profoundly influence human interactions. For example, when handling heavy objects, individuals naturally use a broader palm area than with lighter ones. Considering these intricate human behaviours is vital for achieving improved realism in 3D motion synthesis.

To address the aforementioned challenges, this thesis advances the state of the art of the single view RGB-based 3D human motion capture by introducing the explicit modellings of physics and interactions between the human and the scene, and on the body itself. Furthermore, this thesis proposes the first method for the 3D object manipulation synthesis that

considers the object's physical property (mass) for improved realism and controllability.

1.2 EXPLICIT PHYSICS INTEGRATION

One of the key goals of MoCap methods is to reconstruct the natural and physically plausible 3D human motions that are not visually indistinguishable from real-world motions. This realistic motion can then be used in downstream applications without the need for postprocessing. Many methods have been proposed to capture realistic 3D human motions from lightweight setups such as a monocular camera (Bogo et al., 2016; Chen and Ramanan, 2017; Feng et al., 2021a; Habibie et al., 2019; Kanazawa et al., 2018; Kocabas et al., 2020a; Martinez et al., 2017; Mehta et al., 2017a,b; Moreno-Noguer, 2017; Newell et al., 2016; Pavlakos et al., 2017, 2018b; Rhodin et al., 2018; Sun et al., 2021; Tekin et al., 2016; Tomè et al., 2017; Yang et al., 2018; Zhou et al., 2017). However, due to its highly ill-posed nature, the captured motions can appear unnatural, often due to apparent violations of the laws of physics, such as implausible interpenetrations with the surrounding scene, jitter, unnatural poses, and foot skating. To address these issues, many single-view MoCap methods have attempted to leverage learned priors from 3D motion datasets. However, these approaches cannot entirely prevent many of the aforementioned reconstruction errors as they lack awareness of physics modelling. Chapter 3 proposes an alternative way of mitigating such artefacts in monocular pose estimation algorithms (Shimada et al., 2020). Relying on knowledge from rigid body dynamics, this approach prevents the violation of laws of real-world physics and significantly reduces artefacts in the reconstructed motions. This chapter also introduces a new physical plausibility measurement to assess the reconstructed motion quality from various perspectives. In Chapter 4, a fully learning-based approach is introduced, where the equation of motion is integrated into the method's design (Shimada et al., 2021). The neural network based motion controllers adjust the intensity of the control signals for the humanoid character based on the input motions. This approach yields substantially improved motions compared to optimisation-based MoCap methods (Shimada et al., 2020) in terms of 3D accuracy while faithfully obeying the rigid body dynamics modelling. Moreover, not only are the reconstructed motions improved,

but the estimated contact forces and joint torques also exhibit more plausible values, although they are estimated only from a monocular RGB video, which can be utilised to analyse the motions and assess stress on the human body joints.

1.3 INTERACTIONS WITH THE SCENE

In our daily lives, humans constantly interact with the 3D world through a variety of activities such as walking, sitting, lying, grabbing, and touching. Capturing such motions with the awareness of interactions is of high importance for understanding human behaviour, AR/VR applications, and more. However, capturing motions performed in complex scenes only from a single RGB camera poses substantial challenges (*e.g.* the presence of occlusions caused by objects and frequent interactions with the scene on different body parts such as the back, hands, feet, and buttocks). As a consequence, the reconstructed motions often show inaccurate motions, implausible interactions and collisions with the scene geometry. While numerous scene-aware MoCap works have been proposed (Hassan et al., 2019; Li et al., 2022; Rempe et al., 2021, 2020; Zanfir et al., 2018; Zhang et al., 2021d), yet the aforementioned issues remain unresolved. Chapter 5 addresses the problem of human motion capture with scene interactions from a single RGB input and a scene point cloud (Shimada et al., 2022). The method comprises two innovative components that significantly improve the plausibility of the reconstructed 3D motions. The first component involves predicting the contact regions on both the human body and the scene geometry surfaces using a pixel-aligned implicit function. These estimated contact regions are then integrated into the fitting optimisation process, effectively reducing the inherent depth ambiguity of the single-view setup. The second component of the method resolves severe body-environment collisions. This is achieved by introducing a novel sampling-based optimisation technique in a learned pose prior manifold, significantly resolving these collisions in a hard manner. Thanks to the two novel components, the final reconstructed motion sequences demonstrate significantly more realistic motions interacting with the environment compared to the prior works in the field.

1.4 CAPTURING FACE DEFORMATIONS

Hand-face interactions are a common occurrence in our daily lives, where we touch our faces in various ways, such as poking a cheek, touching the nose, or pinching the chin. During these interactions, the human face surfaces often deform, which conveys facial expressions beyond those produced solely by contracting facial muscles. Capturing these deformation effects, along with the motions of the face and hands, is crucial for applications that require immersive experiences such as VR, avatar communications, etc. However, this aspect has not been addressed so far in MoCap research. This problem casts challenges due to the constant occlusions at the interaction regions and the lack of 3D dataset for training learning approaches. Furthermore, highly accurate 3D localisation of the hand and face is crucial to avoid collisions between them, particularly in the interaction area. This aspect becomes even more challenging when considering a single-view RGB setup due to its inherent depth ambiguity. Chapter 6 addresses this problem and introduces several significant contributions (Shimada et al., 2023). This research marks the first approach to tackle the challenge of capturing deformation in hand-face interactions from a single view. The work proposes the first dataset that contains multiview RGB videos with their corresponding 3D geometries with surface deformations for hand-face interactions. The method consists of the neural networks trained on this dataset. These networks estimate plausible deformations and contacts resulting from the hand-face interactions. Additionally, this chapter introduces learned hand-face interaction priors to enhance the 3D localisation accuracy further. The final reconstructed motions show highly accurate face and hand poses as well as plausible deformations caused by their interactions.

1.5 MASS-AWARE MOTION SYNTHESIS

Our daily interactions with objects are significantly influenced by their physical properties, such as mass. For instance, an object's weight can change how we grasp it and the speed at which we move it. These intricate behavioural changes need to be considered for synthesising 3D object interactions to achieve improved realism. Surprisingly, this aspect has not been addressed by the existing works on simulating hand object

interactions (Christen et al., 2022; Ghosh et al., 2023; Zhang et al., 2021c; Zheng et al., 2023). Chapter 7 tackles this problem by proposing the first mass-conditioned 3D hand and object motion synthesis approach (Shimada et al., 2024). The synthesised 3D hand-object interactions adjust their behaviour based on the object's mass and the interaction type. The method also accepts a user-specified 3D object trajectory as input and synthesises the natural 3D hand motions conditioned by the object's mass. This flexibility allows the method to be used for diverse applications, from generating datasets for machine learning tasks to expediting animation production. The comprehensive experiments verify that the generated interactions from the method are highly realistic and plausible.

1.6 STRUCTURE

This thesis comprises the following eight chapters:

- Chapter 1 provides an overview of the research scope of this thesis, motivating the individual research topics. Additionally, it summarises the structure and outlines the list of publications.
- Chapter 2 elaborates on background concepts needed for the understanding of the thesis.
- Chapter 3 presents the 3D human motion capture algorithm that explicitly integrates physics modelling from rigid body dynamics (Shimada et al., 2020). By introducing the laws of physics, the captured 3D human motions exhibit natural 3D motions suppressing typical artefacts such as unnatural foot skating, irregular poses, spurious body translations, motion jitter and environment collisions.
- Chapter 4 proposes the fully learning-based 3D human motion capture algorithm (Shimada et al., 2021). Unlike the prior MoCap works on the basis of explicit physics laws, the method is fully neural network-based. The components learn the parameters of proportional derivative (PD) controllers that actuate the humanoid character. The networks adjust the intensity of the PD controller signals based on the observed motions (*e.g.* higher intensities for fast motions, and lower intensities for non-fast motions). The re-

constructed 3D motions show improved plausibility and accuracy compared to the prior works.

- Chapter 5 presents a motion capture method that explicitly models the whole body human-environment interactions from an RGB video and corresponding 3D scene represented by a point cloud (Shimada et al., 2022). There are two key innovations to the approach. First, the method estimates dense contact labels on both the body and environment surfaces, which subsequently guide the fitting optimisation for improved 3D localisations. Second, the method introduces a novel sampling algorithm in a learned pose manifold space to resolve the physically implausible collisions in a hard manner. Our experiments show that this work outperforms the prior works by a big margin in terms of 3D accuracy and physical plausibility.
- Chapter 6 describes a new method for single view face-hand interaction capture along with the interaction-induced deformations (Shimada et al., 2023). This chapter proposes a novel dataset capture pipeline that integrates the deformable object simulator — position based dynamics (PBD) — into a markerless multi-view tracking system to obtain the ground truth 3D deformations of the face tissues. Furthermore, the neural network based method estimates the 3D face and hand positions along with the surface deformations while reducing the artefacts compared to other related works.
- Chapter 7 introduces a 3D motion synthesis method for object manipulation with hands, where the object's mass conditions the motion (Shimada et al., 2024). The synthesised 3D motions show highly natural manipulations faithfully reflecting the influence of the object's mass. The method optionally accepts a user-specified object trajectory as input and synthesises the 3D motions that follow the provided trajectory. This functionality can substantially streamline the process of 3D animation creation. This chapter also explains the simple data capture methodology for capturing the hand-object interactions using a multi-view camera setup.
- Chapter 8 concludes this thesis by providing a summary and the insights derived from the preceding chapters. Additionally, this

chapter offers a discussion on open questions remaining to be addressed in the future.

1.7 LIST OF PUBLICATIONS

In the following, the peer-reviewed publications and journals, which are discussed in this thesis are listed:

- Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt (2020). "PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time." In: *ACM Transactions on Graphics (TOG)* 39.6
- Soshi Shimada, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt (2021). "Neural Monocular 3D Human Motion Capture with Physical Awareness." In: *ACM Transactions on Graphics (TOG)* 40.4
- Soshi Shimada, Vladislav Golyanik, Zhi Li, Patrick Pérez, Weipeng Xu, and Christian Theobalt (2022). "HULC: 3D HUman Motion Capture with Pose Manifold SampLing and Dense Contact Guidance." In: *European Conference on Computer Vision (ECCV)*
- Soshi Shimada, Vladislav Golyanik, Patrick Pérez, and Christian Theobalt (2023). "Decaf: Monocular Deformation Capture for Face and Hand Interactions." In: *ACM Transactions on Graphics (TOG)* 42.6
- Soshi Shimada, Franziska Mueller, Jan Bednarik, Bardia Doosti, Bernd Bickel, Danhang Tang, Vladislav Golyanik, Jonathan Taylor, Christian Theobalt, and Thabo Beeler (2024). "MACS: Mass Conditioned 3D Hand and Object Motion Synthesis." In: *International Conference on 3D Vision (3DV)*

Additionally, the peer-reviewed conference and journal publications to which I contributed during my doctoral study are listed in the following:

- Erik C.M. Johnson, Marc Habermann, Soshi Shimada, Vladislav Golyanik, and Christian Theobalt (2023). "Unbiased 4D: Monocular 4D Reconstruction with a Neural Deformation Model." In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*

- Zhi Li, Soshi Shimada, Bernt Schiele, Christian Theobalt, and Vladislav Golyanik (2022). "MoCapDeform: Monocular 3D Human Motion Capture in Deformable Scenes." In: *International Conference on 3D Vision (3DV)*
- Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik (2022). "UnrealEgo: A New Dataset for Robust Egocentric 3D Human Motion Capture." In: *European Conference on Computer Vision (ECCV)*
- Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu (2022). "Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors." In: *Computer Vision and Pattern Recognition (CVPR)*
- Rishabh Dabral, Soshi Shimada, Arjun Jain, Christian Theobalt, and Vladislav Golyanik (2021). "Gravity-Aware Monocular 3D Human-Object Reconstruction." In: *International Conference on Computer Vision (ICCV)*
- Jameel Malik, Soshi Shimada, Ahmed Elhayek, Sk Aziz Ali, Christian Theobalt, Vladislav Golyanik, and Didier Stricker (2021). "Handvoxnet++: 3d hand shape and pose estimation using voxel-based neural networks." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 44.12, pp. 8962–8974
- Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker (2020). "Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map." In: *Computer Vision and Pattern Recognition (CVPR)*
- Vladislav Golyanik, Soshi Shimada, and Christian Theobalt (2020). "Fast simultaneous gravitational alignment of multiple point sets." In: *International Conference on 3D Vision (3DV)*

2

PRELIMINARIES

This chapter introduces several key concepts crucial for understanding the methods presented in this thesis. First, foundational concepts needed for modelling the human body are introduced. Second, detailed descriptions of the fundamentals of rigid body dynamics are provided.

2.1 3D HUMAN BODY REPRESENTATION

Human bodies have a very complex structure: bones, muscles, blood vessels, organs, blood circulations, etc. Therefore, a simplified human 3D model is often employed in computer graphics and vision research. The skeleton of a human body is represented as a tree structure described in Sec. 2.1.1. The explicit surface of a human body is typically represented as a mesh, *i.e.* a set of 3D vertices in a Cartesian coordinate and their edge connection information. Such a high-dimensional representation is often parametrised for better controllability and dimensionality reduction purposes. Sec. 2.1.2 explains the parametric representation of the human body surface.

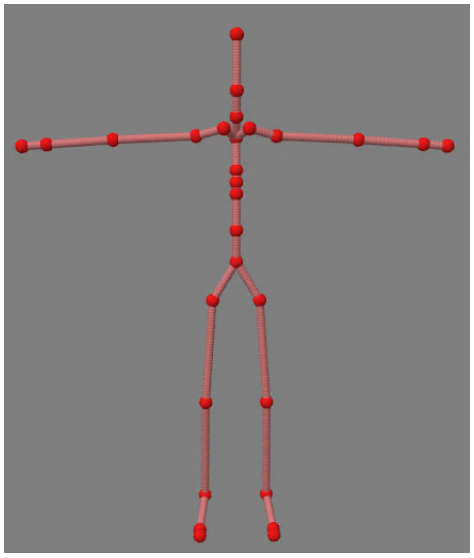


Figure 2.1: Exemplary human skeleton structure. The red dots represent the 3D body joint positions. The edges between the joints are the hypothetical lines that imitate simplified human bones.

2.1.1 *Skeletal representation of human body*

As our human body is highly complex, a 3D articulated human pose is often represented by a set of 3D joint positions with edges that connect

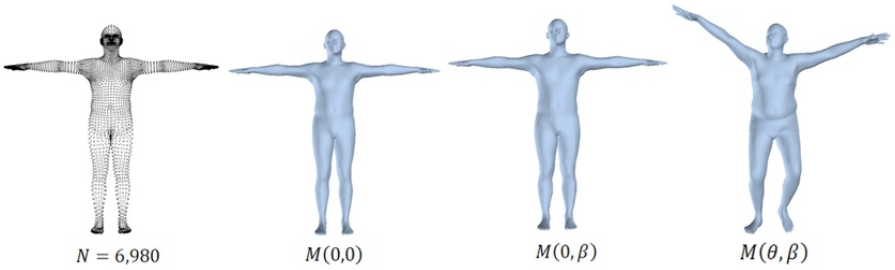


Figure 2.2: Example of PCA-based human body parametric model M. Semantic vectors control the body model; a shape vector $\boldsymbol{\beta}$ for representing various human body shapes and a pose vector $\boldsymbol{\theta}$ for the articulations of the body. The image is taken from Xie et al. (2019).

them; see Fig. 2.1 for an example. The skeleton model is configured by root translation $\boldsymbol{\tau} \in \mathbb{R}^3$ and rotation $\boldsymbol{\phi} \in \mathrm{SO}(3)$, joint angles $\boldsymbol{\theta} \in \mathbb{R}^{3k}$ and bone lengths $\mathbf{l} \in \mathbb{R}^{k-1}$, where k denotes the number of joints. The 3D joint positions $\mathbf{J} \in \mathbb{R}^{3k}$ can be obtained using a forward kinematic function that accepts the skeleton configurations as input.

2.1.2 *Parametric representation of human body*

The surface of a human body is typically represented as a mesh that consists of vertices and edge information. To reduce the high dimension of the surface representation, PCA-based or neural network-based parametric models are often utilised, especially in tasks not specific to individual identity. These parametric models typically accept semantic parameters such as a shape and pose as inputs and return the vertex positions or sign distance field (SDF) of the body surface. In this thesis, PCA-based parametric model for a body (Pavlakos et al., 2019), a face (Li et al., 2017) and hands (Romero et al., 2017) are used due to their simplicity and low time and space complexities, see Fig. 2.2 for the exemplary PCA human model.

2.2 BASIS OF RIGID BODY DYNAMICS

In this subsection, the fundamentals of the rigid body dynamics modelling leveraged in Chapter 3 and 4 are described.

Let $\mathbf{q} \in \mathbb{R}^m$ be the kinematic state of an articulated humanoid character. The first six elements of $\mathbf{q}$ represent the root translation followed by the root rotation of the character. $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ represent the first and second order time derivative of $\mathbf{q}$. Using a finite difference method, the relations of $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are modelled as follows:

$$\begin{aligned} \dot{\mathbf{q}}^{i+1} &= \dot{\mathbf{q}}^{i} + \Delta t \ddot{\mathbf{q}}^{i}, \\ \mathbf{q}^{i+1} &= \mathbf{q}^{i} + \Delta t \dot{\mathbf{q}}^{i+1}, \end{aligned} \tag{2.1}$$

where Δt denotes the time interval, and the superscript i represents the time step. In our physical world, we are able to execute motions by obtaining external forces from the environment and generating internal muscle forces. Based on Newton's second law, the relationship between the force, acceleration and mass of an articulated body is formulated as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T \mathbf{G} \lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}). \tag{2.2}$$

This equation is called equation of motion describing the relations between the mass and inertia of the character represented by a mass matrix $\mathbf{M} \in \mathbb{R}^{m \times m}$, torque vector $\boldsymbol{\tau} \in \mathbb{R}^m$, contact force $\lambda \in \mathcal{R}^{3N_c}$, and the summarised gravity, Coriolis and centripetal forces $\mathbf{c} \in \mathbb{R}^m$. $\mathbf{M}$ contains the distribution of mass and moment of inertia properties of the character. $\boldsymbol{\tau}$ determines the internal joint torque, mimicking the summarised forces of all muscles attached to the respective bones. λ models the N_c contact forces raised from external-interactions. $\mathbf{G}$ is a matrix that converts λ into linear and rotational forces. Without loss of generality, we assume $N_c = 1$. Then the formulation of $\mathbf{G}$ reads:

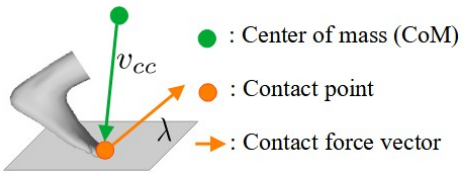


Figure 2.3: Schematic visualisation of the contact force λ and the vector v_{cc} that directs from the centre of mass of the body to the contact point.

$$\mathbf{G} = \begin{bmatrix} E \\ \chi \end{bmatrix} \tag{2.3}$$

where $E \in \mathcal{R}^{3\times3}$ is an identity matrix. χ is an operator that computes a cross-product between the contact force vector λ and a vector $v_{cc} =$

$[v_{cc}^x \; v_{cc}^y \; v_{cc}^z]^T$ that directs from the centre of mass (CoM) of the character to the contact point, see Fig. 2.3 for its schematic visualisation. The operator χ is the cross product rewritten in matrix form as

$$\begin{bmatrix} 0 & -v_{cc}^z & v_{cc}^y \\ v_{cc}^z & 0 & -v_{cc}^x \\ -v_{cc}^y & v_{cc}^x & 0 \end{bmatrix}. \tag{2.4}$$

For the contact forces λ to be physically plausible, the force vectors need to reside in a so-called friction cone. For computational efficiency, the friction cone constraint F is often linearly approximated:

$$F^j = \left\{ \lambda^j \in \mathcal{R}^3 \,\middle|\, \lambda_n^j > 0, \left| \lambda_t^j \right| \leq \bar{\mu}\lambda_n^j, \left| \lambda_b^j \right| \leq \bar{\mu}\lambda_n^j \right\} \tag{2.5}$$

where λ_b and λ_t are the tangential component, and λ_n denotes the normal component of the contact force λ, respectively. $\bar{\mu}$ represents the friction coefficient of an inner linear cone.

2.3 PD CONTROLLER

A proportional derivative (PD) controller is one type of controller frequently used in a control system. The output of the controller is in proportion to the error signal and its derivative. This concept is leveraged in the Chapters 3 and 4. In the context of character control, the PD controller is often used to estimate the actuation force $\boldsymbol{\tau}$ given the target kinematic state $\mathbf{q}_{\text{tar.}}$ and its time derivative $\dot{\mathbf{q}}_{\text{tar.}}$:

$$\boldsymbol{\tau} = \mathbf{k}_p \circ (\mathbf{q}_{\text{tar.}} - \mathbf{q}) + \mathbf{k}_d \circ (\dot{\mathbf{q}}_{\text{tar.}} - \dot{\mathbf{q}}), \tag{2.6}$$

where $\mathbf{k}_p$ and $\mathbf{k}_d$ are coefficient values that weight the first term (error signal) and the second term (derivative of the error signal), respectively. "$\circ$" denotes the Hadamard matrix product.

3

PHYSCAP: PHYSICALLY PLAUSIBLE MONOCULAR 3D MOTION CAPTURE IN REAL TIME

This chapter introduces a novel monocular RGB-based 3D human motion capture method that embeds explicit modelling of physics into the approach (published as Shimada et al., 2020). Unlike the prior works that directly estimate the 3D joint angles of a kinematic skeleton from the video input, the introduced method estimates the forces and accelerations that drive the humanoid character to reconstruct motions. As a result of this in-depth modelling of physics, the reconstructed motions display significantly fewer artefacts (*e.g.* jitters, environment collisions, foot-skating unnatural body leaning, etc). Furthermore, this chapter introduces new metrics to evaluate the plausibility of the reconstructed 3D motions from the perspective of temporal consistency and body-environment collisions.

3.1 INTRODUCTION

3D human pose estimation from monocular RGB images is a very active area of research. Progress is fueled by many applications with an increasing need for reliable, real-time and simple-to-use pose estimation. Here, applications in character animation, VR and AR, telepresence, or human-computer interaction, are only a few examples of high importance for graphics.

Monocular and markerless 3D capture of the human skeleton is a highly challenging and severely underconstrained problem (Kovalenko et al., 2019; Martinez et al., 2017; Mehta et al., 2017b; Pavlakos et al., 2018a; Wandt and Rosenhahn, 2019). Even the best state-of-the-art algorithms, therefore, exhibit notable limitations. Most methods capture pose kinematically using individually predicted joints but do not produce smooth joint angles of a coherent kinematic skeleton. Many approaches perform per-frame pose estimates with notable temporal jitter, and reconstructions are often in root-relative but not global 3D space. Even if a global pose is predicted, depth prediction from the camera is often unstable.

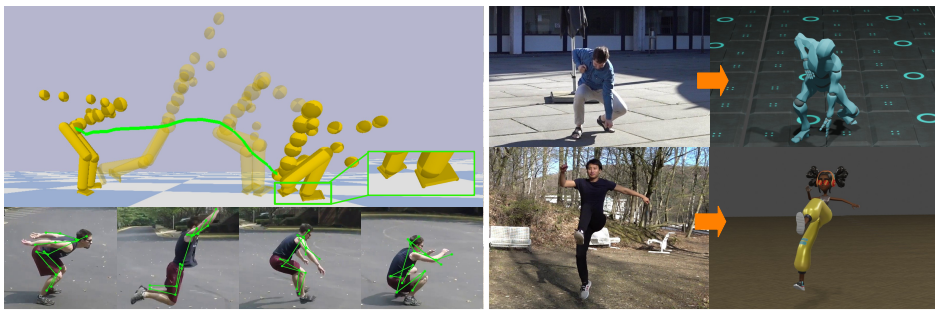


Figure 3.1: *PhysCap* captures global 3D human motion in a physically plausible way from monocular videos in real-time, automatically and without the use of markers. (Left:) Video of a *standing long jump* (Peng et al., 2018b) and our 3D reconstructions with substantially mitigated artefacts, thanks to the formulation on the basis of physics-based dynamics in our method. (Right:) Our *PhysCap* can directly drive virtual characters without any further post-processing. The 3D characters are taken from Adobe (2020).

Also, interaction with the environment is usually entirely ignored, which leads to poses with severe collision violations, *e.g.* floor penetration or the implausible foot sliding and incorrect foot placement. Established kinematic formulations also do not explicitly consider the biomechanical plausibility of reconstructed poses, yielding reconstructed poses with improper balance, inaccurate body leaning, or temporal instability.

We note that all these artefacts are particularly problematic in the aforementioned computer graphics applications, in which temporally stable and visually plausible motion control of characters from all virtual viewpoints, in global 3D, and with respect to the physical environment, are critical. Further on, we note that established metrics in widely-used 3D pose estimation benchmarks (Ionescu et al., 2013; Mehta et al., 2017a), such as mean per joint position error (MPJPE) or 3D percentage of correct keypoints (3D-PCK), which are often even evaluated after a 3D rescaling or Procrustes alignment, do not adequately measure these artefacts. In fact, we show (see Sec. 3.4) that even some top-performing methods on these benchmarks produce results with substantial temporal noise and unstable depth prediction, with frequent violation of environment constraints, and with frequent disregard of physical and anatomical pose plausibility. In consequence, there is still a notable gap between monocular 3D pose human estimation approaches and the gold standard accuracy and motion quality of suit-based or marker-based motion capture systems, which are unfortunately expensive, complex to use and not

suited for many of the aforementioned applications requiring in-the-wild capture.

We, therefore, present *PhysCap* – a new approach for easy-to-use monocular global 3D human motion capture that significantly narrows this gap and substantially reduces the aforementioned artefacts, see Fig. 3.1 for an overview. *PhysCap* is, to our knowledge, the first method that jointly possesses all the following properties: it is fully automatic, markerless, works in general scenes, runs in real-time, captures a space-time coherent skeleton pose and global 3D pose sequence of state-of-the-art temporal stability and smoothness. It exhibits state-of-the-art posture and position accuracy, and captures physically and anatomically plausible poses that correctly adhere to physics and environment constraints. To this end, we rethink and bring together in new way ideas from kinematics-based monocular pose estimation and physics-based human character animation.

The *first stage* of our algorithm is similar to (Mehta et al., 2017b) and estimates 3D body poses in a purely kinematic, physics-agnostic way. A convolutional neural network (CNN) infers combined 2D and 3D joint positions from an input video, which are then refined in space-time inverse kinematics to yield the first estimate of skeletal joint angles and global 3D poses. In the *second stage*, the foot contact and the motion states are predicted for every frame. Therefore, we employ a new CNN that detects heel and forefoot placement on the ground from estimated 2D keypoints in images, and classifies the observed poses into stationary or non-stationary. In the *third stage*, the final physically plausible 3D skeletal joint angle and pose sequence is computed in real-time. This stage regularises human motion with a torque-controlled physics-based character represented by a kinematic chain with a floating base. To this end, the optimal control forces for each degree of freedom (DoF) of the kinematic chain are computed, such that the kinematic pose estimates from the first stage – in both 2D and 3D – are reproduced as closely as possible. The optimisation ensures that physics constraints like gravity, collisions, foot placement, as well as physical pose plausibility (*e.g.* balancing), are fulfilled. To summarise, our **contributions** in this chapter are:

- The first, to the best of our knowledge, marker-less monocular 3D human motion capture approach on the basis of an explicit physics-

based dynamics model which runs in real-time and captures global, physically plausible skeletal motion (Sec. 3.4).

- A CNN to detect foot contact and motion states from images (Sec. 3.4.2).
- A new pose optimisation framework with a human parametrised by a torque-controlled simulated character with a floating base and PD joint controllers; it reproduces kinematically captured 2D/3D poses and simultaneously accounts for physics constraints like ground reaction forces, foot contact states and collision response (Sec. 3.4.3).
- Quantitative metrics to assess frame-to-frame jitter and floor penetration in captured motions (Sec. 3.5.3.1).
- Physically-justified results with significantly fewer artefacts, such as frame-to-frame jitter, incorrect leaning, foot sliding and floor penetration than related methods (confirmed by a user study and metrics), as well as state-of-the-art 2D and 3D accuracy and temporal stability (Sec. 3.5).

We demonstrate the benefits of our approach through the experimental evaluations on several datasets (including newly recorded videos) against multiple state-of-the-art methods for monocular 3D human motion capture and pose estimation.

3.2 RELATED WORK

Our method mainly relates to two different categories of approaches – (markerless) 3D human motion capture from colour imagery, and physics-based character animation. In the following, we review related types of methods, focusing on the most closely related works.

3.2.1 *Multi-View RGB Methods for 3D Human MoCap*

Reconstructing humans from multi-view images is well-studied. Multi-view motion capture methods track the articulated skeletal motion, usually by fitting an articulated template to imagery (Bo and Sminchisescu, 2008; Brox et al., 2010; Elhayek et al., 2016, 2014; Gall et al., 2010; Stoll

et al., 2011; Wang et al., 2018; Zhang et al., 2020c). Other methods, sometimes termed performance capture methods, additionally capture the non-rigid surface deformation, *e.g.* of clothing (Cagniart et al., 2010; Starck and Hilton, 2007; Vlasic et al., 2009; Waschbüsch et al., 2005). They usually fit some form of a template model to multi-view imagery (Bradley et al., 2008; De Aguiar et al., 2008; Martin-Brualla et al., 2018) that often also has an underlying kinematic skeleton (Gall et al., 2009; Liu et al., 2011; Vlasic et al., 2008; Wu et al., 2012). Multi-view methods have demonstrated compelling results and some enable free-viewpoint video. However, they require expensive multi-camera setups and often controlled studio environments.

3.2.2 *Monocular RGB 3D Human MoCap and Pose Estimation*

Marker-less 3D human pose estimation (reconstruction of 3D joint positions only) and motion capture (reconstruction of global 3D body motion and joint angles of a coherent skeleton) from a single colour or greyscale image are highly ill-posed problems. The state of the art on monocular 3D human pose estimation has greatly progressed in recent years, mostly fueled by the power of trained CNNs (Habibie et al., 2019; Mehta et al., 2017a). Some methods estimate 3D pose by combining 2D keypoints prediction with body depth regression (Dabral et al., 2018; Newell et al., 2016; Yang et al., 2018; Zhou et al., 2017) or with regression of 3D joint location probabilities (Mehta et al., 2017b; Pavlakos et al., 2017) in a trained CNN. Lifting methods predict joint depths from detected 2D keypoints (Chen and Ramanan, 2017; Martinez et al., 2017; Pavlakos et al., 2018a; Tomè et al., 2017). Other CNNs regress 3D joint locations directly (Mehta et al., 2017a; Rhodin et al., 2018; Tekin et al., 2016). Another category of methods combines CNN-based keypoint detection with constraints from a parametric body model, *e.g.* by using reprojection losses during training (Bogo et al., 2016; Brau and Jiang, 2016; Habibie et al., 2019). Some works approach monocular multi-person 3D pose estimation (Rogez et al., 2019) and motion capture (Mehta et al., 2020), or estimate non-rigidly deforming human surface geometry from monocular video on top of skeletal motion (Habermann et al., 2020, 2019; Xu et al., 2020). In addition to greyscale images, Xu et al. (2020) use an

asynchronous event stream from an event camera as input. Both these latter directions are complementary but orthogonal to our work.

The majority of methods in this domain estimate 3D pose as a root-relative 3D position of the body joints (Kovalenko et al., 2019; Martinez et al., 2017; Moreno-Noguer, 2017; Pavlakos et al., 2018a; Wandt and Rosenhahn, 2019). This is problematic for applications in graphics, as temporal jitter, varying bone lengths and the often not recovered global 3D pose make animating virtual characters hard. Other monocular methods are trained to estimate parameters or joint angles of a skeleton (Zhou et al., 2016) or parametric model (Kanazawa et al., 2018). Mehta et al. (2020, 2017b) employ inverse kinematics on top of CNN-based 2D/3D inference to obtain joint angles of a coherent skeleton in global 3D and in real-time.

Results of all aforementioned methods frequently violate laws of physics, and exhibit foot-floor penetrations, foot sliding, and unbalanced or implausible poses floating in the air, as well as notable jitter. Some methods try to reduce jitter by exploiting temporal information (Kanazawa et al., 2019; Kocabas et al., 2020a), *e.g.* by estimating smooth multi-frame scene trajectories (Peng et al., 2018b). Zou et al. (2020) try to reduce foot sliding by ground contact constraints. Zanfir et al. (2018) jointly reason about ground planes and volumetric occupancy for multi-person pose estimation. Monszpart et al. (2019) jointly infer coarse scene layout and human pose from monocular interaction video, and Hassan et al. (2019) use a pre-scanned 3D model of scene geometry to constrain kinematic pose optimisation. No prior motion capture works formulate explicit physics-based modelling and real-time capability, unlike ours.

3.2.3 *Physics-Based Character Animation*

Character animation with physics-based controllers has been investigated for many years (Barzel et al., 1996; Sharon and Panne, 2005; Wrotek et al., 2006), and remains an active area of research, (Andrews et al., 2016; Bergamin et al., 2019; Levine and Popović, 2012; Zheng and Yamane, 2013). Levine and Popović (2012) employ a quasi-physical simulation that approximates a reference motion trajectory in real time. They can follow non-physical reference motion by applying a direct actuation at the root. By using proportional derivative (PD) controllers and computing optimal

torques and contact forces, Zheng and Yamane (2013) make a character follow a reference motion captured while keeping balance. Liu et al. (2010) proposed a probabilistic algorithm for physics-based character animation. Due to the stochastic property and inherent randomness, their results evince variations, but the method requires multiple minutes of runtime per sequence. Andrews et al. (2016) employ rigid dynamics to drive a virtual character from a combination of marker-based motion capture and body-mounted sensors. This animation setting is related to motion transfer onto robots. Nakaoka et al. (2007) transferred human motion captured by a multi-camera marker-based system onto a robot, with an emphasis on leg motion. Zhang et al. (2014) leverage depth cameras and wearable pressure sensors and apply physics-based motion optimisation. We take inspiration from these works for our setting, where we have to capture in a physically correct way and in real-time global 3D human motion from images, using intermediate pose reconstruction results that exhibit notable artefacts and violations of physics laws. *PhysCap*, therefore, combines an initial kinematics-based pose reconstruction with PD controller based physical pose optimisation.

Several recent methods apply deep reinforcement learning to virtual character animation control (Bergamin et al., 2019; Lee et al., 2019; Peng et al., 2018b). Peng et al. (2018b) propose a reinforcement learning approach for transferring dynamic human performances observed in monocular videos. They first estimate smooth motion trajectories with recent monocular human pose estimation techniques, and then train an imitating control policy for a virtual character. Bergamin et al. (2019) train a controller for a virtual character from several minutes of motion capture data, which covers the expected variety of motions and poses. Once trained, the virtual character can follow the directional commands of the user in real time, while being robust to collisional obstacles. Other work (Lee et al., 2019) combines a muscle actuation model with deep reinforcement learning. Jiang et al. (2019) express an animation objective in muscle actuation space. The work on learning animation controllers for specific motion classes is inspirational but different from real-time physics-based motion capture of general motion.

3.2.4 *Physically Plausible Monocular 3D Human Motion Capture*

Only a few works on monocular 3D human motion capture using explicit physics-based constraints exist (Li et al., 2019; Vondrak et al., 2012; Wei and Chai, 2010; Zell et al., 2017). Wei and Chai (2010) capture 3D human poses from uncalibrated monocular video using physics constraints. Their approach requires manual user input for each frame of a video. In contrast, our approach is automatic, runs in real time, and uses a different formulation for physics-based pose optimisation geared to our setting. Vondrak et al. (2012) capture bipedal controllers from a video. Their controllers are robust to perturbations and generalise well for a variety of motions. However, unlike our *PhysCap*, the generated motion often looks unnatural and their method does not run in real time. Zell et al. (2017) capture poses and internal body forces from images only for certain classes of motion (*e.g.* lifting and walking) by using a data-driven approach, but not an explicit forward dynamics approach handling a wide range of motions, like ours.

Our *PhysCap* bears most similarities with the rigid body dynamics based monocular human pose estimation by Li et al. (2019). They estimate 3D poses, contact states and forces from input videos with physics-based constraints. However, their method and our approach are substantially different. While Li et al. (2019) focus on object-person interactions, we target a variety of general motions, including complex acrobatic motions such as backflipping without objects. Their method does not run in real-time and requires manual annotations on images to train the contact state estimation networks. In contrast, we leverage the PD controller based inverse dynamics tracking, which results in physically plausible, smooth and natural skeletal pose and root motion capture in real-time. Moreover, our contact state estimation network relies on annotations generated in a semi-automatic way. This enables our architecture to be trained on large datasets, which results in improved generalisability. No previous method of the reviewed category "physically plausible monocular 3D human motion capture" combines *the ability of our algorithm to capture global 3D human pose of similar quality and physical plausibility in real time*.

3.3 BODY MODEL AND PRELIMINARIES

The input to *PhysCap* is a 2D image sequence $\mathbf{I}_t$, $t \in \{1, \ldots, T\}$, where T is the total number of frames and t is the frame index. We assume a perspective camera model and calibrate the camera and floor location before tracking starts. Our approach outputs a physically plausible real-time 3D motion capture result $\mathbf{q}^t_{phys} \in \mathbb{R}^m$ (where m is the number of degrees of freedom) that adheres to the image observation, as well as physics-based posture and environment constraints. For our human model, $m = 43$. Joint angles are parametrised by Euler angles. The mass distribution of our character is computed following Liu et al. (2010). Our character model has a skeleton composed of 37 joints and *links*. A link defines the volumetric extent of a body part via a collision proxy. The forefoot and heel links, centred at the respective joints of our character (see Fig. 3.2), are used to detect foot-floor collisions during physics-based pose optimisation.

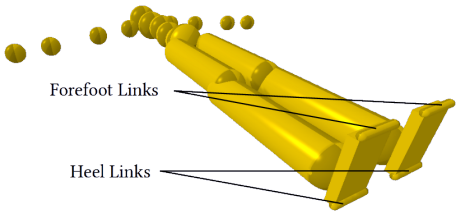


Figure 3.2: Our virtual character used in stage III. The forefoot and heel links are involved in the mesh collision checks with the floor plane in the physics engine (Coumans and Bai, 2016).

Throughout our algorithm, we represent the pose of our character by a combined vector $\mathbf{q} \in \mathbb{R}^m$ (Featherstone, 2014). The first three entries of $\mathbf{q}$ contain the global 3D root position in Cartesian coordinates, the next three entries encode the orientation of the root, and the remaining entries are the joint angles. When solving for the physics-based motion capture result, the motion of the physics-based character will be controlled by the vector of forces denoted by $\boldsymbol{\tau} \in \mathbb{R}^m$ interacting with gravity, Coriolis and centripetal forces $\mathbf{c} \in \mathbb{R}^m$. The root of our character is not fixed and can globally move in the environment, which is commonly called a floating-base system. Let the velocity and acceleration of $\mathbf{q}$ be $\dot{\mathbf{q}} \in \mathbb{R}^m$ and $\ddot{\mathbf{q}} \in \mathbb{R}^m$, respectively. Using the finite-difference method, the relationship between $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ can be written as

$$\begin{aligned} \dot{\mathbf{q}}^{i+1} &= \dot{\mathbf{q}}^i + \phi \ddot{\mathbf{q}}^i, \\ \mathbf{q}^{i+1} &= \mathbf{q}^i + \phi \dot{\mathbf{q}}^{i+1}, \end{aligned} \tag{3.1}$$

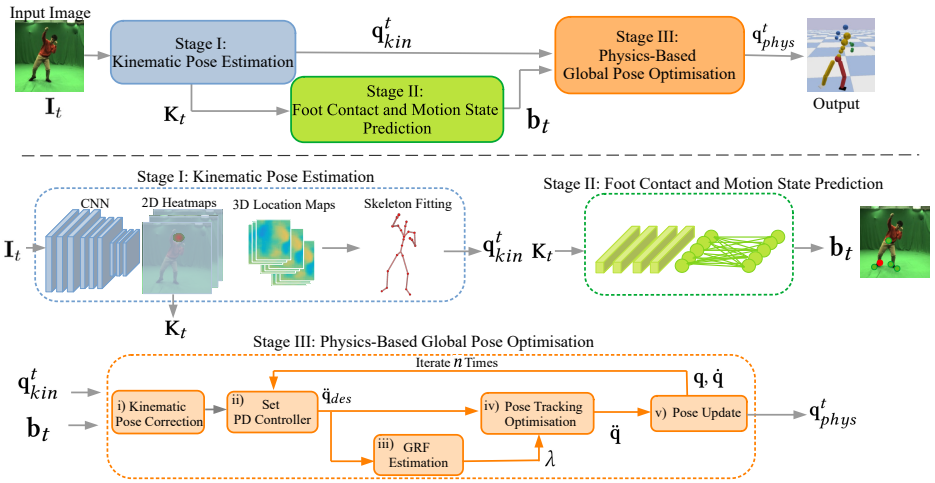


Figure 3.3: Overview of our pipeline.

where i represents the simulation step index and $\phi = 0.01$ is the simulation step size. For the motion to be physically plausible, $\ddot{\mathbf{q}}$ and the vector of forces $\boldsymbol{\tau}$ must satisfy the equation of motion (Featherstone, 2014):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T\mathbf{G}\lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \tag{3.2}$$

where $\mathbf{M} \in \mathbb{R}^{m \times m}$ is a joint space inertia matrix which is composed of the moment of inertia of the system. It is computed using the Composite Rigid Body algorithm (Featherstone, 2014). $\mathbf{J} \in \mathbb{R}^{6N_c \times m}$ is a contact Jacobi matrix which relates the external forces to joint coordinates, with N_c denoting the number of links where the contact force is applied. $\mathbf{G} \in \mathcal{R}^{6N_c \times 3N_c}$ transforms contact forces $\lambda \in \mathcal{R}^{3N_c}$ into the linear force and torque (Zheng and Yamane, 2013).

Usually, in a floating-base system, the first six entries of $\boldsymbol{\tau}$ which correspond to the root motion are set to 0 for a humanoid character control. This reflects the fact that humans do not directly control root translation and orientation by muscles acting on the root, but indirectly by the other joints and muscles in the body. In our case, however, the kinematic pose $\mathbf{q}^t_{kin}$ which our final physically plausible result shall reproduce as much as possible (see Sec. 3.4), is estimated from a monocular image sequence (see stage I in Fig. 3.3), which contains artefacts. Solving for joint torque controls that blindly make the character follow, would make the character quickly fall down. Hence, we keep the first six entries of $\boldsymbol{\tau}$ in our formulation and can thus directly control the root position and orientation

with an additional external force. This enables the final character motion to keep up with the global root trajectory estimated in the first stage of *PhysCap*, without falling down.

3.4 METHOD

Our *PhysCap* approach includes three stages, see Fig. 3.3 for an overview. The first stage performs *kinematic pose estimation*. This encompasses 2D heatmap and 3D location map regression for each body joint with a CNN, followed by a model-based space-time pose optimisation step (Sec. 3.4.1). This stage returns 3D skeleton pose in joint angles $\mathbf{q}_{kin}^t \in \mathbb{R}^m$ along with the 2D joint keypoints $\mathbf{K}_t \in \mathbb{R}^{s\times 2}$ for every image; s denotes the number of 2D joint keypoints. As explained earlier, this initial kinematic reconstruction $\mathbf{q}_{kin}^t$ is prone to physically implausible effects such as foot-floor penetration, foot skating, anatomically implausible body leaning and temporal jitter, especially notable along the depth dimension.

The second stage performs *foot contact and motion state detection*, which uses 2D joint detections $\mathbf{K}_t$ to classify the poses reconstructed so far into stationary and non-stationary – this is stored in one binary flag. It also estimates binary foot-floor contact flags, *i.e.* for the toes and heels of both feet, resulting in four binary flags (Sec. 3.4.2). This stage outputs the combined state vector $\mathbf{b}_t \in \mathbb{R}^5$.

The third and final stage of *PhysCap* is the *physically plausible global 3D pose estimation* (Sec. 3.4.3). It combines the estimates from the first two stages with physics-based constraints to yield a physically plausible real-time 3D motion capture result that adheres to physics-based posture and environment constraints $\mathbf{q}_{phys}^t \in \mathbb{R}^m$. In the following, we describe each of the stages in detail.

3.4.1 *Stage I: Kinematic Pose Estimation*

Our kinematic pose estimation stage follows the real-time VNect algorithm (Mehta et al., 2017b), see Fig. 3.3, stage I. We first predict heatmaps of 2D joints and root-relative location maps of joint positions in 3D with a specially tailored fully convolutional neural network using ResNet (He et al., 2016). The ground truth joint locations for training are taken from

the MPII (Andriluka et al., 2014) and LSP (Johnson and Everingham, 2011) datasets in the 2D case, and MPI-INF-3DHP (Mehta et al., 2017a) and Human3.6m (Ionescu et al., 2013) datasets in the 3D case.

Next, the estimated 2D and 3D joint locations are temporally filtered and used as constraints in a kinematic skeleton fitting step that optimises the following energy function:

$$\begin{aligned}\mathbf{E}_{kin}(\mathbf{q}_{kin}^t) =& \mathbf{E}_{\text{IK}}(\mathbf{q}_{kin}^t) + \mathbf{E}_{\text{proj.}}(\mathbf{q}_{kin}^t) + \\ & \mathbf{E}_{\text{smooth}}(\mathbf{q}_{kin}^t) + \mathbf{E}_{\text{depth}}(\mathbf{q}_{kin}^t).\end{aligned} \tag{3.3}$$

The energy function (3.3) contains four terms (see Mehta et al. (2017b)), *i.e.* the 3D inverse kinematics term $\mathbf{E}_{\text{IK}}$, the projection term $\mathbf{E}_{\text{proj.}}$, the temporal stability term $\mathbf{E}_{\text{smooth}}$ and the depth uncertainty correction term $\mathbf{E}_{\text{depth}}$. $\mathbf{E}_{\text{IK}}$ is the data term which constrains the 3D pose to be close to the 3D joint predictions from the CNN. $\mathbf{E}_{\text{proj.}}$ enforces the pose $\mathbf{q}_{kin}^t$ to reproject it to the 2D keypoints (joints) detected by the CNN. Note that this reprojection constraint, together with calibrated camera and calibrated bone lengths, enables computation of the global 3D root (pelvis) position in the camera space. Temporal stability is further imposed by penalising the root's acceleration and variations along the depth channel by $\mathbf{E}_{\text{smooth}}$ and $\mathbf{E}_{\text{depth}}$, respectively. The energy (3.3) is optimised by Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963), and the obtained vector of joint angles and the root rotation and position $\mathbf{q}_{kin}^t$ of a skeleton with fixed bone lengths are smoothed by an adaptive first-order low-pass filter (Casiez et al., 2012). Skeleton bone lengths of a human can be computed, up to a global scale, from averaged 3D joint detections of a few initial frames. Knowing the metric height of the human determines the scale factor to compute metrically correct global 3D poses.

The result of stage I is a temporally consistent joint angle sequence but, as noted earlier, captured poses can exhibit artefacts and contradict physical plausibility (*e.g.* evince floor penetration, incorrect body leaning, temporal jitter, *etc.*).

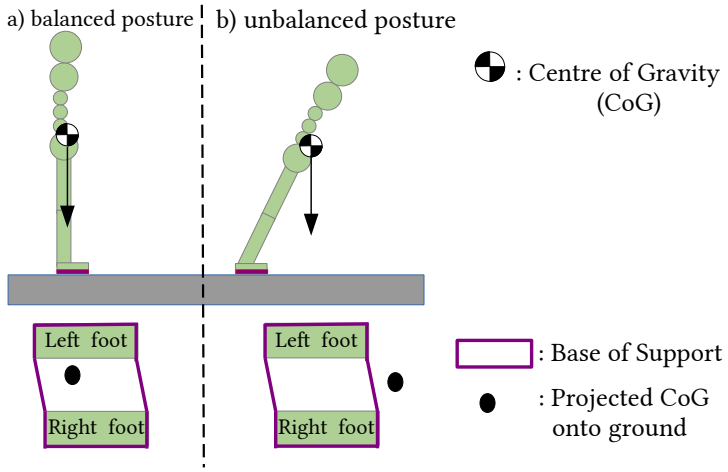


Figure 3.4: (a) Balanced posture: the CoG of the body projects inside the base of support. (b) Unbalanced posture: the CoG does not project inside the base of support, which causes the human to start losing balance.

3.4.2 *Stage II: Foot Contact and Motion State Detection*

The ground reaction force (GRF) – applied when the feet touch the ground – enables humans to walk and control their posture. The interplay of internal body forces and the ground reaction force controls the human pose, which enables locomotion and body balancing by controlling the centre of gravity (CoG). To compute physically plausible poses accounting for the GRF in stage III, we thus need to know foot-floor contact states. Another important aspect of the physical plausibility of biped poses, in general, is balance. When a human is standing or in a stationary upright state, the CoG of the body projects inside a base of support (BoS). The BoS is an area on the ground bounded by the foot contact points, see Fig. 3.4 for a visualisation. When the CoG projects outside the BoS in a stationary pose, a human starts losing balance and will fall if no correcting motion or step is applied. Therefore, maintaining a static pose with an extensive leaning, as often observed in the results of monocular pose estimation, is not physically plausible (Fig .3.4-(b)). The aforementioned CoG projection criterion can be used to correct imbalanced stationary poses (Coros et al., 2010; Faloutsos et al., 2001; Macchietto et al., 2009). To perform such correction in stage III, we need to know if a pose is stationary or non-stationary (whether it is a part of a locomotion/walking phase).

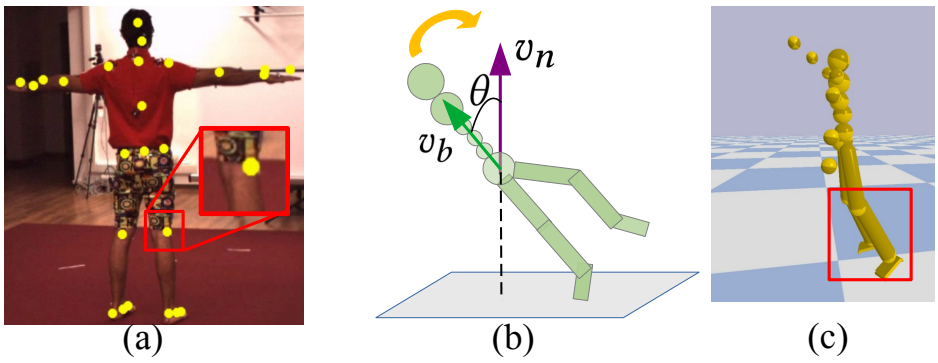


Figure 3.5: (a) An exemplary frame from the Human 3.6M dataset with the ground truth reprojections of the 3D joint keypoints. The magnified view in the red rectangle shows the reprojected keypoint that deviates from the rotation centre (the middle of the knee). (b) Schematic visualisation of the reference motion correction. Readers are referred to Sec. 3.4.3.1 for its details. (c) Example of a visually unnatural standing (stationary) pose caused by physically implausible knee bending.

Stage II, therefore, estimates foot-floor contact states of the feet in each frame and determines whether the pose of the subject in $\mathbf{I}_t$ is stationary or not. To predict both, *i.e.* foot contact and motion states, we use a neural network whose architecture extends Zou et al. (2020) who only predict foot contacts. It is composed of temporal convolutional layers with one fully connected layer at the end. The network takes as input all 2D keypoints $\mathbf{K}_t$ from the last seven time steps (the temporal window size is set to seven), and returns for each image frame binary labels indicating whether the subject is in the stationary or non-stationary pose, as well as the contact state flags for the forefeet and heels of both feet encompassed in $\mathbf{b}_t$. The supervisory labels for training this network are automatically computed on a subset of the 3D motion sequences of the Human3.6M (Ionescu et al., 2013) and DeepCap (Habermann et al., 2020) datasets using the following criteria: the forefoot and heel joint contact labels are computed based on the assumption that a joint in contact is not sliding, *i.e.* the velocity is lower than 5 cm/sec. In addition, we use a height criterion, *i.e.* the forefoot/heel, when in contact with the floor, has to be at a 3D height that is lower than a threshold $h_{\text{thres.}}$. To determine this threshold for each sequence, we calculate the average heel h^{heel}_{avg} and forefoot h^{ffoot}_{avg} heights for each subject using the first ten frames (when both feet touch the ground). Thresholds are then computed as $h^{heel}_{\text{thres.}} = h^{heel}_{\text{avg}} + 5\text{cm}$ for heels and $h^{ffoot}_{\text{thres.}} = h^{ffoot}_{\text{avg}} + 5\text{cm}$ for the forefeet.

This second criterion is needed since, otherwise, a foot in the air that is kept static could also be labelled as being in contact.

We also automatically label stationary and non-stationary poses on the same sequences. When standing and walking, the CoG of the human body typically lies close to the pelvis in 3D, which corresponds to the skeletal root position in both the Human3.6M and DeepCap datasets. Therefore, when the velocity of 3D root is lower than a threshold φ_v, we classify the pose as *stationary*, and *non-stationary* otherwise. In total, around $600k$ sets of contact and motion state labels for the human images are generated.

3.4.3 *Stage III: Physically Plausible Global 3D Pose Estimation*

Stage III uses the results of stages I and II as inputs, *i.e.* $\mathbf{q}_{kin}^t$ and $\boldsymbol{b}_t$. It transforms the kinematic motion estimate into a physically plausible global 3D pose sequence that corresponds to the images and adheres to anatomy and environmental constraints imposed by the laws of physics. To this end, we represent the human as a torque-controlled simulated character with a floating base and PD joint controllers (A. Salem and Aly, 2015). The core is to solve an energy-based optimisation problem to find the vector of forces $\boldsymbol{\tau}$ and accelerations $\ddot{\mathbf{q}}$ of the character such that the equations of motion with constraints are fulfilled (Sec. 3.4.3.5). This optimisation is preceded by several preprocessing steps applied to each frame.

First i), we correct $\mathbf{q}_{kin}^t$ if it is strongly implausible based on several easy-to-test criteria (Sec. 3.4.3.1). Second ii), we estimate the desired acceleration $\ddot{\mathbf{q}}_{des} \in \mathbb{R}^m$ necessary to reproduce $\mathbf{q}_{kin}^t$ based on the PD control rule (Secs. 3.4.3.2). Third iii), in input frames in which a foot is in contact with the floor (Sec. 3.4.3.3), we estimate the ground reaction force (GRF) λ (Sec. 3.4.3.4). Fourth iv), we solve the optimisation problem (3.9) to estimate $\boldsymbol{\tau}$ and accelerations $\ddot{\mathbf{q}}$ where the equation of motion with the estimated GRF λ and the contact constraint to avoid foot-floor penetration (Sec. 3.4.3.5) are integrated as constraints. Note that the contact constraint is integrated only when the foot is in contact with the floor. Otherwise, only the equation of motion without GRF is introduced as a constraint in (3.9). v) Lastly, the pose is updated using the finite-difference method

(Eq. (3.1)) with the estimated acceleration $\ddot{\mathbf{q}}$. The steps ii) - v) are iterated $n = 4$ times for each frame of video.

As also observed by Andrews et al. (2016), this two-step optimisation iii) and iv) reduces direct actuation of the character's root as much as possible (which could otherwise lead to slightly unnatural locomotion), and explains the kinematically estimated root position and orientation by torques applied to other joints as much as possible when there is a foot-floor contact. Moreover, this two-step optimisation is computationally less expensive rather than estimating $\ddot{\mathbf{q}}$, $\boldsymbol{\tau}$ and λ simultaneously (Zheng and Yamane, 2013). Our algorithm thus finds a plausible balance between pose accuracy, physical accuracy, the naturalness of captured motion and real-time performance.

3.4.3.1 *Pose Correction*

Due to the error accumulation in stage I (*e.g.* as a result of the deviation of 3D annotations from the joint rotation centres in the skeleton model, see Fig. 3.5-(a), as well as inaccuracies in the neural network predictions and skeleton fitting), the estimated 3D pose $\mathbf{q}^t_{kin}$ is often not physically plausible. Therefore, prior to torque-based optimisation, we pre-correct a pose $\mathbf{q}^t_{kin}$ from stage I if it is 1) stationary and 2) unbalanced, *i.e.* the CoG projects outside the base of support (BoS). If both correction criteria are fulfilled, we compute the angle θ_t between the ground plane normal v_n and the vector v_b that defines the direction of the spine relative to the root in the local character's coordinate system (see Fig. 3.5-(b) for the schematic visualisation). We then correct the orientation of the virtual character towards a posture, for which CoG projects inside BoS. Correcting θ_t in one large step could lead to instabilities in physics-based pose optimisation. Instead, we reduce θ_t by a small rotation of the virtual character around its horizontal axis (*i.e.* the axis passing through the transverse plane of a human body) starting with the corrective angle $\xi_t = \frac{\theta_t}{10}$ for the first frame. Thereby, we accumulate the degree of correction in ξ for the subsequent frames, *i.e.* $\xi_{t+1} = \xi_t + \frac{\theta_t}{10}$. Note that θ_t is decreasing for every frame and the correction step is performed for all subsequent frames until 1) the pose becomes non-stationary or 2) CoG projects inside BoS[1].

1 either after the correction or already in $\mathbf{q}^t_{kin}$ provided by stage I

However, simply correcting the spine orientation by the skeleton rotation around the horizontal axis can lead to implausible standing poses, since the knees can still be unnaturally bent for the obtained upright posture (see Fig. 3.5-(c) for an example). To account for that, we adjust the respective DoFs of the knees and hips such that the relative orientation between the upper legs and spine, as well as the upper and lower legs, are more straight. The hip and knee correction starts if both correction criteria are *still* fulfilled and θ_t is *already* very small. Similarly to the θ correction, we introduce accumulator variables for every knee and every hip. The correction step for knees and hips is likewise performed until 1) the pose becomes non-stationary or 2) CoG projects inside BoS[1].

3.4.3.2 *Computing the Desired Accelerations*

To control the physics-based virtual character such that it reproduces the kinematic estimate $\mathbf{q}_{kin}^t$, we set the desired joint acceleration $\ddot{\mathbf{q}}_{des}$ following the PD controller rule:

$$\ddot{\mathbf{q}}_{des} = \ddot{\mathbf{q}}_{kin}^t + k_p(\mathbf{q}_{kin}^t - \mathbf{q}) + k_d(\dot{\mathbf{q}}_{kin}^t - \dot{\mathbf{q}}). \tag{3.4}$$

The desired acceleration $\ddot{\mathbf{q}}_{des}$ is later used in the GRF estimation step (Sec. 3.4.3.4) and the final pose optimisation (Sec. 3.4.3.5). Controlling the character motion on the basis of a PD controller in the system enables the character to exert torques $\boldsymbol{\tau}$ which reproduce the kinematic estimate $\mathbf{q}_{kin}^t$ while significantly mitigating undesired effects such as joint and base position jitter.

3.4.3.3 *Foot-Floor Collision Detection*

To avoid foot-floor penetration in the final pose sequence and to mitigate contact position sliding, we integrate hard constraints in the physics-based pose optimisation that enforce zero velocity of the forefoot and heel links in Sec. 3.4.3.5. However, these constraints can lead to unnatural motion in rare cases when the state prediction network may fail to estimate the correct foot contact states (*e.g.* when the foot suddenly stops

in the air while walking). We thus update the contact state output of the state prediction network $\mathbf{b}_{t,j\in\{1,\dots,4\}}$, to yield $\mathbf{b}'_{t,j\in\{1,\dots,4\}}$ as follows:

$$\mathbf{b}'_{t,j\in\{1,\dots,4\}} = \begin{cases} 1, & \text{if (} \mathbf{b}^j = 1 \text{ and } h^j < \psi \text{) or} \\ & \text{the } j\text{-th link collides with the floor plane,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

This means we consider a forefoot or heel link to be in contact only if its height h^j is less than a threshold $\psi = 0.1$m above the calibrated ground plane.

In addition, we employ the *Pybullet* (Coumans and Bai, 2016) physics engine to detect foot-floor collision for the left and right foot links. Note that combining the mesh collision information with the predictions from the state prediction network is necessary because 1) the foot may not touch the floor plane in the simulation when the subject's foot is actually in contact with the floor due to the inaccuracy of $\mathbf{q}^t_{kin}$, and 2) the foot can penetrate into the mesh floor plane if the network misdetects the contact state when there is actually a foot contact in $\mathbf{I}_t$.

3.4.3.4 *Ground Reaction Force (GRF) Estimation*

We first compute the GRF λ – when there is a contact between a foot and floor – which best explains the motion of the root as coming from stage I. However, the target trajectory from stage I can be physically implausible, and we will thus eventually also require a *residual* force directly applied on the root to explain the target trajectory; this force will be computed in the final optimisation. To compute the GRF, we solve the following minimisation problem:

$$\begin{aligned} \min_{\lambda} & \|\mathbf{M}_1\ddot{\mathbf{q}}_{des} + \mathbf{c}_1(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{J}_1^T\mathbf{G}\lambda\|, \\ & \text{s.t. } \lambda \in F, \end{aligned} \tag{3.6}$$

where $\|\cdot\|$ denotes ℓ^2-norm, and $\mathbf{M}_1 \in \mathcal{R}^{6\times m}$ together with $\mathbf{J}_1^T \in \mathcal{R}^{6\times 6N_c}$ are the first six rows of $\mathbf{M}$ and $\mathbf{J}^T$ that correspond to the root joint, respectively. $\mathbf{c}_1 \in \mathcal{R}^6$ denotes the first six elements of $\mathbf{c}$ (see Eq.3.2), which also correspond to the root joint. Also, see Chapter 2 for the details of the linearised friction cone constraint F.

The GRF λ is then integrated into the subsequent optimisation step (3.9) to estimate torques and accelerations of all joints in the body, including an additional residual direct root actuation component that is needed to explain the difference between the global 3D root trajectory of the kinematic estimate and the final physically correct result. The aim is to keep this direct root actuation as small as possible, which is best achieved by a two-stage strategy that first estimates the GRF separately. Moreover, we observed this two-step optimisation enables faster computation than estimating λ, $\ddot{\mathbf{q}}$ and $\boldsymbol{\tau}$ all at once. It is, hence, more suitable for our approach that aims at fast operation.

3.4.3.5 *Physics-Based Pose Optimisation*

In this step, we solve an optimisation problem to estimate $\boldsymbol{\tau}$ and $\ddot{\mathbf{q}}$ to track $\mathbf{q}^t_{kin}$ using the equation of motion (3.2) as a constraint. When contact is detected (Sec. 3.4.3.3), we integrate the estimated ground reaction force λ (Sec. 3.4.3.4) in the equation of motion. In addition, we introduce contact constraints to prevent foot-floor penetration and foot sliding when contacts are detected.

Let $\dot{\mathbf{r}}_j$ be the velocity of the j-th contact link. Then, using the relationship between $\dot{\mathbf{r}}_j$ and $\dot{\mathbf{q}}$ (Featherstone, 2014), we can write:

$$\mathbf{J}_j \dot{\mathbf{q}} = \dot{\mathbf{r}}_j. \tag{3.7}$$

When the link is in contact with the floor, the velocity perpendicular to the floor has to be zero or positive to prevent penetration. Also, we allow the contact links to have a small tangential velocity σ to prevent an immediate foot motion stop which creates visually unnatural motion. Our contact constraint inequalities read:

$$0 \leq \dot{r}^n_j, \quad |\dot{r}^t_j| \leq \sigma, \quad \text{and} \quad |\dot{r}^b_j| \leq \sigma, \tag{3.8}$$

where $\dot{r}^n_j$ is the normal component of $\dot{\mathbf{r}}_j$, and $\dot{r}^t_j$ along with $\dot{r}^b_j$ are the tangential elements of $\dot{\mathbf{r}}_j$.

Using the desired acceleration $\ddot{\mathbf{q}}_{des}$ (Eq. (3.4)), the equation of motion (3.2), optimal GRF λ estimated in (3.6) and contact constraints (3.8), we

Table 3.1: Names and duration of our six newly recorded outdoor sequences captured using SONY DSC-RX0 at 25 fps.

Sequence ID	Sequence Name	Duration [sec]
1	*building 1*	132
2	*building 2*	90
3	*forest*	105
4	*backyard*	60
5	*balance beam 1*	21
6	*balance beam 2*	12

formulate the optimisation problem for finding the physics-based motion capture result as:

$$\begin{aligned} &\min_{\ddot{\mathbf{q}},\boldsymbol{\tau}} \|\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{des}\| + \|\boldsymbol{\tau}\|, \\ \text{s.t. } &\mathbf{M}\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^T\mathbf{G}\lambda - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}), \text{ and} \\ &0 \leq \dot{r}_j^n,\ |\dot{r}_j^t| \leq \sigma,\ |\dot{r}_j^b| \leq \sigma, \forall j. \end{aligned} \tag{3.9}$$

The first energy term forces the character to reproduce $\mathbf{q}_{kin}^t$. The second energy term is the regulariser that minimises $\boldsymbol{\tau}$ to prevent overshooting, thus modelling natural human-like motion. After solving (3.9), the character pose is updated by Eq. (3.1). We iterate the steps ii) - v) (see stage III in Fig. 3.3) $n = 4$ times, and stage III returns the n-th output from v) as the final character pose $\mathbf{q}_{phys}^t$. The final output of stage III is a sequence of joint angles and global root translations and rotations that explains the image observations, follows the purely kinematic reconstruction from stage I, yet is physically and anatomically plausible and temporally stable.

3.5 RESULTS

We first provide implementation details of *PhysCap* (Sec. 3.5.1) and then demonstrate its qualitative state-of-the-art results (Sec. 3.5.2). We next evaluate *PhysCap*'s performance quantitatively (Sec. 3.5.3) and conduct a user study to assess the visual physical plausibility of the results (Sec. 3.5.4).

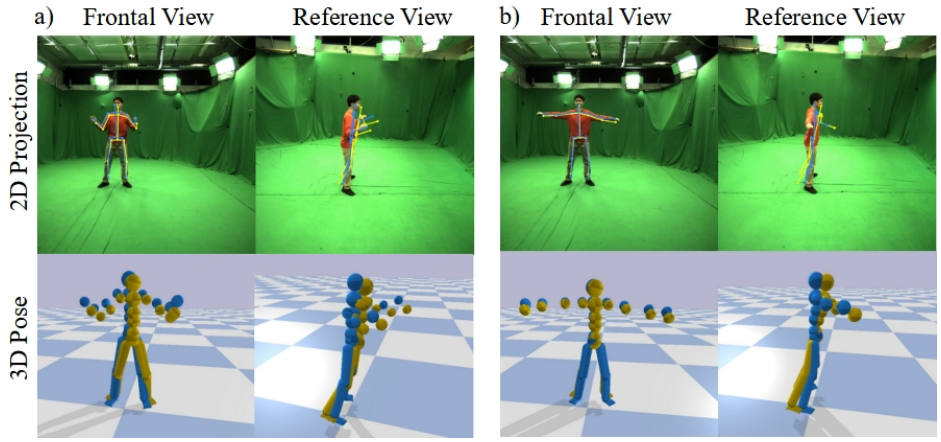


Figure 3.6: Two examples of reprojected 3D keypoints obtained by our approach (light blue colour) and Vnect (Mehta et al., 2017b) (yellow colour) together with the corresponding 3D visualisations from different view angles. *PhysCap* produces much more natural and physically plausible postures, whereas Vnect suffers from unnatural body leaning.

We test *PhysCap* on widely-used benchmarks (Habermann et al., 2020; Ionescu et al., 2013; Mehta et al., 2017a) as well as on *backflip* and *jump* sequences provided by Peng et al. (2018b). We also collect a new dataset with various challenging motions. It features six sequences in general scenes performed by two subjects recorded at 25 fps. For the recording, we used SONY DSC-RX0, see Table 3.1 for more details on the sequences.

3.5.1 *Implementation*

Our method runs in real time (25 fps on average) on a PC with a Ryzen7 2700 8-Core Processor, 32 GB RAM and GeForce RTX 2070 graphics card. In stage I, we proceed from a freely available demo version of VNect (Mehta et al., 2017b). Stages II and III are implemented in *python*. In stage II, the network is implemented with *PyTorch* (Paszke et al., 2019). In stage III, we use the *Rigid Body Dynamics Library* (Felis, 2017) to compute dynamic quantities. We employ the *Pybullet* (Coumans and Bai, 2016) as a physics engine for the character motion visualisation and collision detection. In this work, we set the proportional gain value kp and derivative gain value kd for all joints to 300 and 20, respectively. For the root angular acceleration, kp and kd are set to 340 and 30, respectively. kp

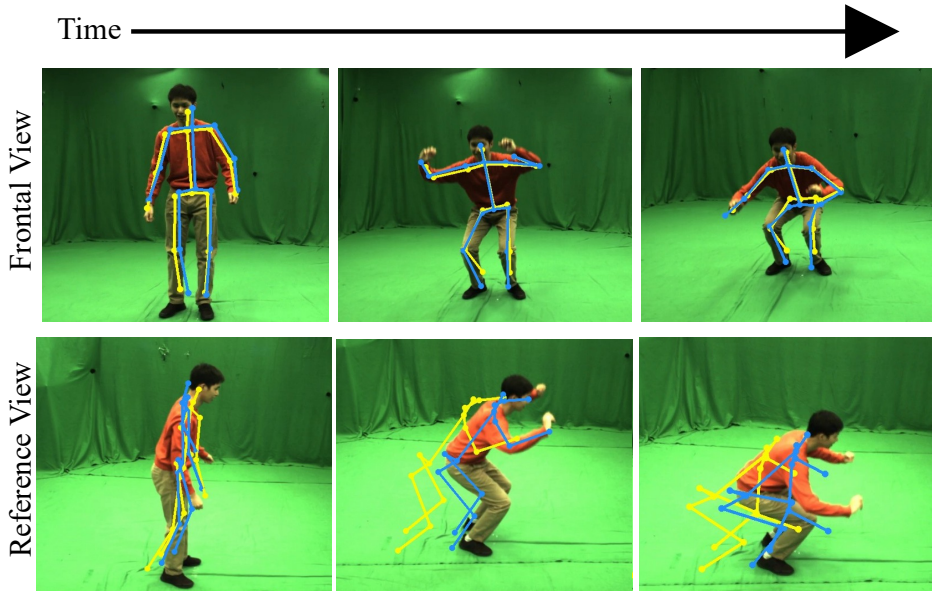


Figure 3.7: Reprojected 3D keypoints onto two different images with different view angles for squatting. Frontal view images are used as inputs, and images of the reference view are used only for quantitative evaluation. Our results are drawn in light blue, whereas the results by VNect (Mehta et al., 2017b) are provided in yellow. Our reprojections are more feasible, which is especially noticeable in the reference view.

and *kd* of the root linear acceleration are set to 1000 and 80, respectively. These settings are used in all experiments.

3.5.2 *Qualitative Evaluation*

Figs. 3.1 and 3.11 show that *PhysCap* captures global 3D human poses in real time, even of fast and difficult motions, such as a backflip and a jump, which are of significantly improved quality compared to previous monocular methods. In particular, captured motions are much more temporally stable, and adhere to laws of physics with respect to the naturalness of body postures and fulfilment of environmental constraints, see Figs. 3.6–3.8 and 3.10 for the examples of more natural 3D reconstructions. These properties are essential for many applications in graphics, in particular for stable real-time character animation, which is feasible by directly applying our method's output (see Fig. 3.1).

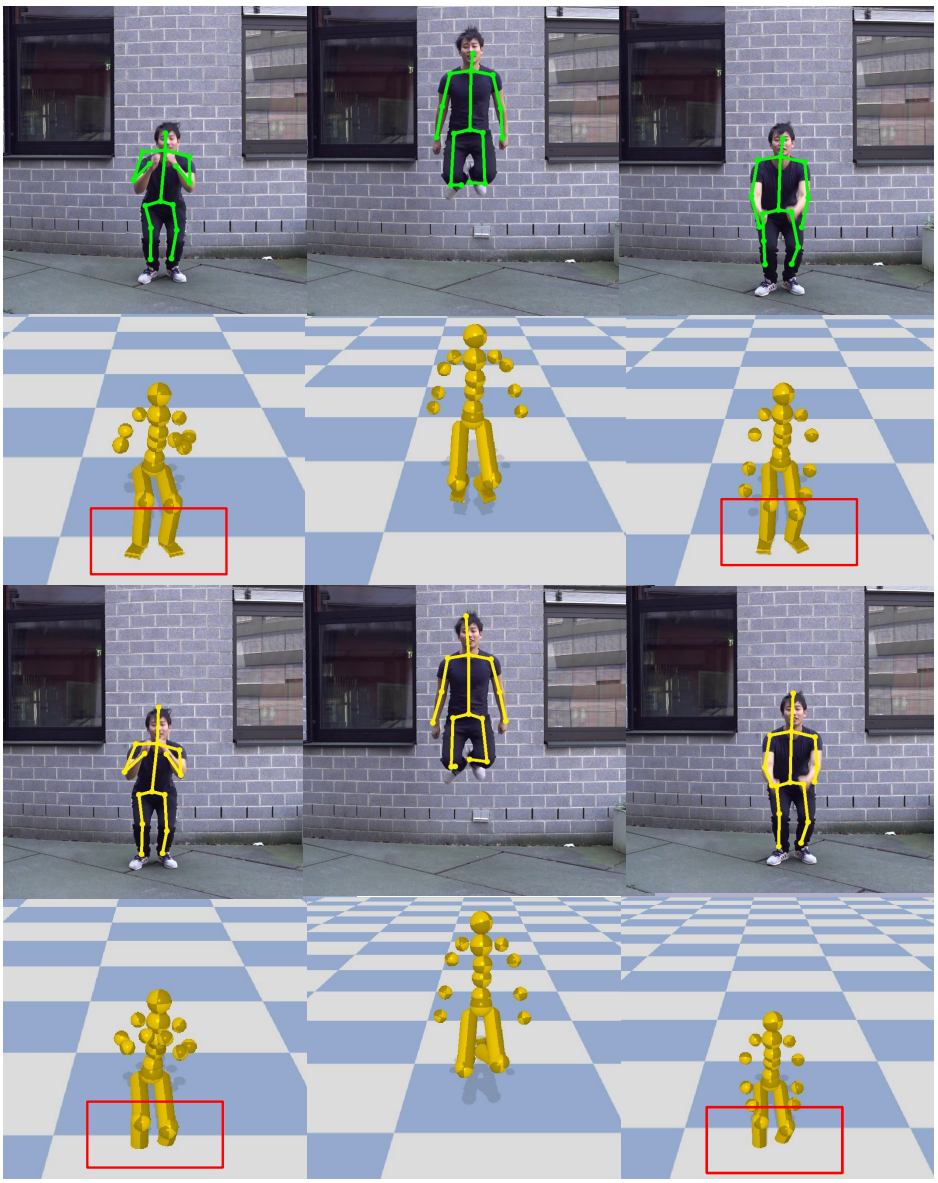


Figure 3.8: Several visualisations of the results by our approach and VNect (Mehta et al., 2017b). The first and second rows show our estimated 3D poses after reprojection in the input image and its 3D view, respectively. Similarly, the third and fourth rows show the reprojected 3D pose and 3D view for VNect. Note that our motion capture shows no foot penetration into the floor plane whereas such an artefact is apparent in the VNect results.

3.5.3 *Quantitative Evaluation*

In the following, we first describe our evaluation method in Sec. 3.5.3.1. We evaluate *PhysCap* and competing methods under a variety of criteria,

Table 3.2: 3D error comparison on benchmark datasets. We report the MPJPE in mm, PCK at 150 mm and AUC. Higher AUC and PCK are better, and lower MPJPE is better. Note that the global root positions for HMR and HMMR were estimated by solving optimisation with a 2D projection loss using the 2D and 3D keypoints obtained from the methods.

		DeepCap			Human 3.6M			MPI-INF-3DHP		
		MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]	MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]	MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]
Procrustes	ours	68.9	**95.0**	57.9	65.1	94.8	60.6	104.4	83.9	43.1
	Vnect	**68.4**	94.9	**58.3**	62.7	95.7	61.9	104.5	84.1	43.2
	HMR	77.1	93.8	52.4	**54.3**	96.9	**66.6**	**87.8**	**87.1**	**50.9**
	HMMR	75.5	93.8	53.1	55.0	**96.6**	66.2	106.9	79.5	44.8
no Procrustes	ours	113.0	75.4	39.3	97.4	82.3	46.4	122.9	72.1	35.0
	Vnect	102.4	80.2	**42.4**	89.6	85.1	49.0	**120.2**	**74.0**	**36.1**
	HMR	113.4	75.1	39.0	**78.9**	88.2	**54.1**	130.5	69.7	35.7
	HMMR	**101.4**	**81.0**	42.0	79.4	**88.4**	53.8	174.8	60.4	30.8
global root position	ours	**110.5**	**80.4**	**37.0**	**182.6**	**54.7**	**26.8**	**257.0**	**29.7**	**15.3**
	Vnect	112.6	80.0	36.8	185.1	54.1	26.5	261.0	28.8	15.0
	HMR	251.4	19.5	8.4	204.2	45.8	22.1	505.0	28.6	13.5
	HMMR	213.0	27.7	11.3	231.1	41.6	19.4	926.2	28.0	14.5

i.e. 3D joint position, reprojected 2D joint positions, foot penetration into the floor plane and motion jitter. We compare our approach with current state-of-the-art monocular pose estimation methods, *i.e.* HMR (Kanazawa et al., 2018), HMMR (Kanazawa et al., 2019) and Vnect (Mehta et al., 2017b) (here we use the so-called *demo version* provided by the authors with further improved accuracy over the original paper due to improved training). For the comparison, we use the benchmark dataset Human3.6M (Ionescu et al., 2013), the DeepCap dataset (Habermann et al., 2020) and MPI-INF-3DHP (Mehta et al., 2017a). From the Human3.6M dataset, we use the subset of actions that does not have occluding objects in the frame, *i.e. directions, discussions, eating, greeting, posing, purchases, taking photos, waiting, walking, walking dog* and *walking together*. From the DeepCap dataset, we use the subject 2 for this comparison.

3.5.3.1 *Evaluation Methodology*

The established evaluation methodology in monocular 3D human pose estimation and capture consists of testing a method on multiple sequences and reporting the accuracy of 3D joint positions as well as the accuracy of the reprojection into the input views. The accuracy in 3D is evaluated

Table 3.3: 2D projection error of a frontal view (input) and side view (non-input) on DeepCap dataset (Habermann et al., 2020). *PhysCap* performs similarly to VNect on the frontal view, and significantly better on the side view. For further details, see Sec. 3.5.3 and Fig. 3.7.

	Front View		Side View	
	e_{2D}^{input} [pixel]	$\sigma_{2D}^{\text{input}}$	e_{2D}^{side} [pixel]	$\sigma_{2D}^{\text{side}}$
Ours	21.1	6.7	**35.5**	**16.8**
Vnect (Mehta et al., 2017b)	**14.3**	**2.7**	37.2	18.1

by *mean per joint position error* (MPJPE) in mm, *percentage of correct keypoints* (PCK) and the *area under the receiver operating characteristic (ROC) curve* abbreviated as AUC. The reprojection or mean pixel error e_{2D}^{input} is obtained by projecting the estimated 3D joints onto the input images and taking the average per frame distance to the ground truth 2D joint positions. We report e_{2D}^{input} and its standard deviation denoted by $\sigma_{2D}^{\text{input}}$ with the images of size 1024×1024 pixels.

As explained earlier, these metrics only evaluate limited aspects of captured 3D poses and do not account for essential aspects of temporal stability, smoothness and physical plausibility in reconstructions such as jitter, foot sliding, foot-floor penetration and unnaturally balanced postures. Moreover, MPJPE and PCK are often reported after rescaling of the result in 3D or Procrustes alignment, which further makes these metrics agnostic to the aforementioned artefacts. Thus, we introduce four additional metrics which allow to evaluate the physical plausibility of the results, *i.e. reprojection error to unseen views* e_{2D}^{side}, *motion jitter error* e_{smooth} and two floor penetration errors – *Mean Penetration Error (MPE)* and *Percentage of Non-Penetration (PNP)*.

When choosing a reference side view for e_{2D}^{side}, we make sure that the viewing angle between the input and side views has to be sufficiently large, *i.e.* more than $\sim\frac{\pi}{15}$. Otherwise, if a side view is close to the input view, such effects as unnatural leaning forward can still remain undetected by e_{2D}^{side} in some cases. After reprojection of a 3D structure to an image plane of a side view, all further steps for calculating e_{2D}^{side} are similar to the steps for the standard reprojection error. We also report $\sigma_{2D}^{\text{side}}$, *i.e.* the standard deviation of e_{2D}^{side}.

Table 3.4: Comparison of temporal smoothness on the DeepCap (Habermann et al., 2020) and Human 3.6M datasets (Ionescu et al., 2013). *PhysCap* significantly outperforms VNect and HMR, and fares comparably to HMMR in terms of this metric. For a detailed explanation, see Sec. 3.5.3.

		Ours	Vnect	HMR	HMMR
DeepCap	e_{smooth}	**6.3**	11.6	11.7	8.1
	σ_{smooth}	**4.1**	8.6	9.0	5.1
Human 3.6M	e_{smooth}	7.2	11.2	11.2	**6.8**
	σ_{smooth}	6.9	10.1	12.7	**5.9**

To quantitatively compare the motion jitter, we report the deviation of the temporal consistency from the ground truth 3D pose. Our smoothness error e_{smooth} is computed as follows:

$$\begin{aligned} Jit_X &= \|\mathbf{p}_X^{s,t} - \mathbf{p}_X^{s,t-1}\|, \\ Jit_{GT} &= \|\mathbf{p}_{GT}^{s,t} - \mathbf{p}_{GT}^{s,t-1}\|, \\ e_{smooth} &= \frac{1}{Tm}\sum_{t=1}^{T}\sum_{s=1}^{m}|Jit_{GT} - Jit_X|, \end{aligned} \tag{3.10}$$

where $\mathbf{p}^{s,t}$ represents the 3D position of joint s in the time frame t. T and m denote the total numbers of frames in the video sequence and target 3D joints, respectively. The subscripts X and GT stand for the predicted output and ground truth, respectively. A lower e_{smooth} indicates lower motion jitter in the predicted motion sequence.

MPE and PNP measure the degree of non-physical foot penetration into the ground. MPE is the mean distance between the floor and 3D foot position, and it is computed only when the foot is in contact with the floor. We use the ground truth foot contact labels (Sec. 3.4.2) to judge the presence of the actual foot contacts. The complementary PNP metric shows the ratio of frames where the feet are not below the floor plane over the entire sequence.

3.5.3.2 *Quantitative Evaluation Results*

Table 3.2 summarises MPJPE, PCK and AUC for root-relative joint positions with (first row) and without (second row) Procrustes alignment before the error computation for our and related methods. We also report the global root position accuracy in the third row. Since HMR and HMMR

Table 3.5: Comparison of Mean Penetration Error (MPE) and Percentage of Non-Penetration (PNP) on DeepCap dataset (Habermann et al., 2020). *PhysCap* significantly outperforms VNect on this metric, measuring an essential aspect of physical motion correctness.

	MPE [mm] ↓	σ_{MPE} ↓	PNP [%] ↑
Ours	**28.0**	**25.9**	**92.9**
Vnect (Mehta et al., 2017b)	39.3	37.5	45.6

do not return global root positions as their outputs, we estimate the root translation in 3D by solving an optimisation with 2D projection energy term using the 2D and 3D keypoints obtained from these algorithms (similar to the solution in VNect). The 3D bone lengths of HMR and HMMR were rescaled so that they match the ground truth bone lengths.

In terms of MPJPE, PCK and AUC, our method does not outperform the other approaches consistently but achieves an accuracy that is comparable and often close to the highest on Human3.6M, DeepCap and MPI-INF-3DHP. In the third row, we additionally evaluate the global 3D base position accuracy, which is critical for character animation from the captured data. Here, *PhysCap* consistently outperforms the other methods on all the datasets.

As noted earlier, the above metrics only paint an incomplete picture. Therefore, we also measure the 2D projection errors to the input and side views on the DeepCap dataset, since this dataset includes multiple synchronised views of dynamic scenes with a wide baseline. Table 3.3 summarises the mean pixel errors e_{2D}^{input} and e_{2D}^{side} together with their standard deviations. In the frontal view, *i.e.* on e_{2D}^{input}, VNect has higher accuracy than *PhysCap*. However, this comes at the prize of frequently violating physics constraints (floor penetration) and producing unnaturally leaning and jittering 3D poses. In contrast, since *PhysCap* explicitly models physical pose plausibility, it excels VNect in the side view, which reveals VNect's implausibly leaning postures and root position instability in depth, also see Figs. 3.6 and 3.7.

To assess motion smoothness, we report e_{smooth} and its standard deviation σ_{smooth} in Table 3.4. Our approach outperforms Vnect and HMR by a big margin on both datasets. Our method is better than HMMR on DeepCap dataset and marginally worse on Human3.6M. HMMR has an explicit temporal component in the architecture.

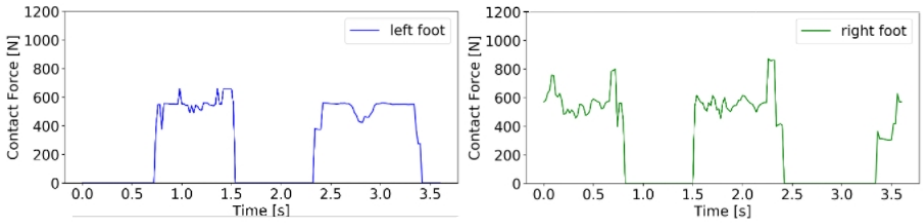


Figure 3.9: The estimated contact forces as the functions of time for the walking sequence. We observe that the contact forces remain in a reasonable range for walking motions (Shahabpoor and Pavic, 2017).

Table 3.5 summarises the MPE and PNP for Vnect and *PhysCap* on DeepCap dataset. Our method shows significantly better results compared to VNect, *i.e.* about a 30% lower MPE and a by 100% better result in PNP, see Fig. 3.8 for qualitative examples. Fig. 3.9 shows plots of contact forces as the functions of time calculated by our approach on the walking sequence from our newly recorded dataset (sequence 1). The estimated functions fall into a reasonable force range for walking motions (Shahabpoor and Pavic, 2017).

3.5.4 *User Study*

The notion of *physical plausibility* can be understood and perceived subjectively from person to person. Therefore, in addition to the quantitative evaluation with existing and new metrics, we perform an online user study which allows to subjectively assess and compare the perceived degree of different effects in the reconstructions by a broad audience of people with different backgrounds in computer graphics and vision. In total, we prepared 34 questions with videos, in which we always showed one or two reconstructions at a time (our result, a result by a competing method, or both at the same time). In total, 27 respondents have participated.

There were different types of questions. In 16 questions (category I), the respondents were asked to decide which 3D reconstruction out of two looks more physically plausible to them (the first, the second or undecided). In 12 questions (category II), the respondents were asked to rate how natural the 3D reconstructed motions are or evaluate the degree of an indicated effect (foot sliding, body leaning, *etc.*) on a predefined

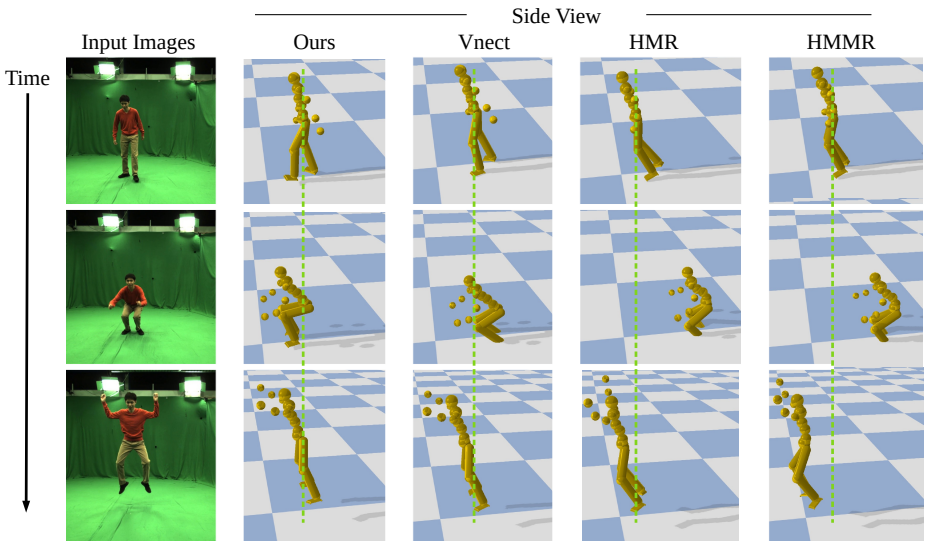


Figure 3.10: Several side (non-input) view visualisations of the results by our approach, Vnect (Mehta et al., 2017b), HMR (Kanazawa et al., 2018) and HMMR (Kanazawa et al., 2019) on DeepCap dataset. The green dashed lines indicate the expected root positions over time. It is apparent from the side view that our *PhysCap* does not suffer from the unnatural body sliding along the depth direction, unlike other approaches. The global base positions for HMR and HMMR were computed by us using the root-relative predictions of these techniques, see Sec. 3.5.3.2 for more details.

scale. In five questions (category III), the respondents were also asked to decide which visualisation has a more pronounced indicated artefact. For two questions out of five, 2D projections onto the input 2D image sequence were shown, whereas the remaining questions in this category featured 3D reconstructions. Finally (category IV), the participants were encouraged to list which artefacts in the reconstructions seem to be most apparent and most frequent.

In category I, our reconstructions were preferred in 89.2% of the cases, whereas a competing method was preferred in 1.6% of the cases. Note that at the same time, the decision between the methods has not been made in 8.9% cases. In category II, the respondents have also found the results of our approach to be significantly more physically plausible than the results of competing methods. The latter were also found to have consistently more jitter, foot sliding and unnatural body leaning. In category III, noteworthy is also that the participants have indicated a higher average perceived accuracy of our reprojections, *i.e.* 32.7% voted that our results reproject better, whereas the choice felt on the competing methods in

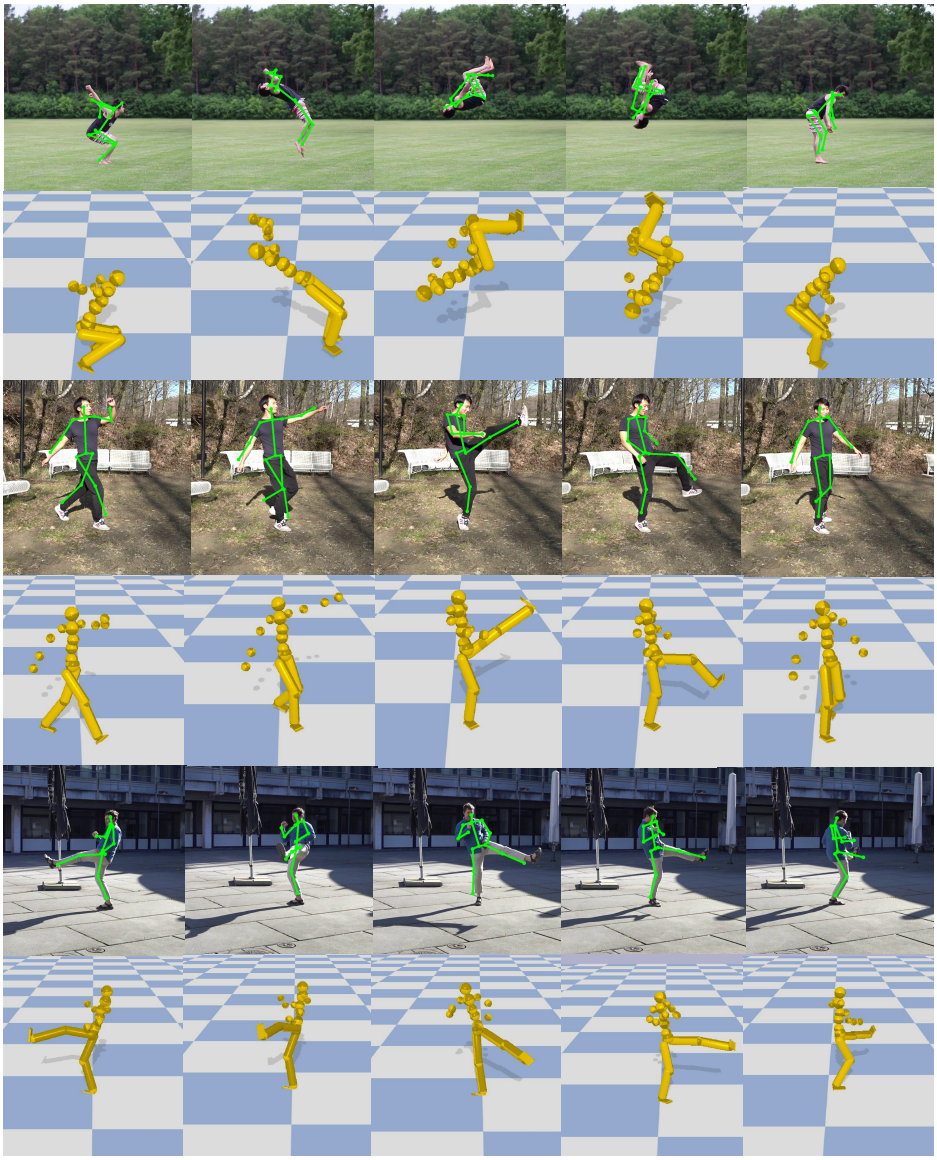


Figure 3.11: Representative 2D reprojections and the corresponding 3D poses of our *PhysCap* approach. Note that, even with the challenging motions, our global poses in 3D have high quality and 2D reprojections to the input images are accurate as well. The *backflip* video in the first row is taken from Peng et al. (2018b). Other sequences are from our own recordings.

22.6% of the cases. Note that the smoothness and jitter in the results are also reflected in the reprojections, and, thus, both influence how natural the reprojected skeletons look like. At the same time, a high uncertainty of 44.2% indicates that the difference between the reprojections of *PhysCap* and other methods is volatile. For the 3D motions in this category, 82.7% voted that our results show fewer indicated artefacts compared to other approaches, whereas 13.5% of the respondents preferred the competing methods. The decision has not been made in 3.7% of the cases. In category IV, 59% of the participants named jitter as the most frequent and apparent disturbing effect of the competing methods, followed by unnatural body leaning (22%), foot-floor penetration (15%) and foot sliding (15%).

The user study confirms a high level of physical plausibility and naturalness of *PhysCap* results. We see that also subjectively, a broad audience coherently finds our results of high visual quality, and the gap to the competing methods is substantial. This strengthens our belief about the suitability of *PhysCap* for computer graphics and primarily virtual character animation in real time.

3.6 DISCUSSION

Our physics-based monocular 3D human motion capture algorithm significantly reduces the common artefacts of other monocular 3D pose estimation methods such as motion jitter, penetration into the floor, foot sliding and unnatural body leaning. The experiments have shown that our state prediction network generalises well across scenes with different backgrounds (see Fig. 3.11). However, in the case of foot occlusion, our state prediction network can sometimes mispredict the foot contact states, resulting in the erroneous hard zero velocity constraint for feet. Additionally, our approach requires the calibrated floor plane to apply the foot contact constraint effectively; standard calibration techniques can be used for this. Swift motions can be challenging for stage I of our pipeline, which can cause inaccuracies in the estimates of the subsequent stages, as well as in the final estimate. In future, other monocular kinematic pose estimators than Mehta et al. (2017b) could be tested in stage I, in case they are trained to handle occlusions and very fast motions better. Moreover, note that – although we use a single parameter set for *PhysCap* in all our experiments (see Sec. 3.5) – users can adjust the quality of the

reconstructed motions by tuning the gain parameters of PD controller depending on the scenario. By increasing the derivative gain value, the reconstructed poses are smoother, which, however, can cause motion delay compared to the input video, especially when the observed motions are very fast. By reducing the derivative gain value, our optimisation with a virtual character can track image sequence with less motion delay, at the cost of less temporally coherent motion.

Further, while our method works in front of general backgrounds, we assume there is a ground plane in the scene, which is the case for most man-made environments, but not irregular outdoor terrains. Finally, our method currently only considers a subset of potential body-to-environment contacts in a physics-based way.

3.7 CONCLUSIONS

We have presented *PhysCap* – the first physics-based approach for a global 3D human motion capture from a single RGB camera that runs in real time at 25 fps. Thanks to the pose optimisation framework using PD joint control, the results of *PhysCap* evince improved physical plausibility, temporal consistency and significantly fewer artefacts such as jitter, foot sliding, unnatural body leaning and foot-floor penetration, compared to other existing approaches (some of them include temporal constraints). We also introduced new error metrics to evaluate these improved properties which are not easily captured by metrics used in the established pose estimation benchmarks. Moreover, our user study further confirmed these improvements.

4

NEURAL MONOCULAR 3D HUMAN MOTION CAPTURE WITH PHYSICAL AWARENESS

The previous chapter introduced a new approach for physics-based 3D human motion capture. This chapter (published as Shimada et al., 2021) introduces a fully learning-based approach for monocular RGB 3D human motion capture with explicit physics modelling. The neural network based character controller is trained with rigid body dynamics modelling, realising the networks to be aware of dynamics and physics quantities. The learned controller adjusts the signal intensity depending on the characteristics of the input video (*e.g.* high intensity when the input contains fast motions such as dancing). Consequently, our new method shows improved 3D accuracy compared with the existing physics-based method proposed in the previous chapter, especially for challenging swift motions. It also surpasses other kinematic-based state-of-the-art methods in terms of the plausibility of the motions. In addition, the predicted ground reaction force and torques show reasonable values, even though they are estimated solely from a video input, which can be helpful for several downstream applications such as sports analysis, rehabilitation, comfort measurement for furniture designs and more.

4.1 INTRODUCTION

3D human motion capture is an actively researched area enabling many applications ranging from human activity recognition to sports analysis, virtual-character animation, film production, human-computer interaction and mixed reality. Since marker-based and multi-camera-based solutions are expensive and unsuited for many applications (*e.g.* in-the-wild capture and recordings outside the studio or legacy content), methods for *markerless* 3D human motion capture from a monocular camera are intensively researched (Mehta et al., 2017b; Shimada et al., 2020).

Monocular 3D human motion capture is a highly challenging inverse problem due to the fundamental ambiguities in deducing 3D body config-

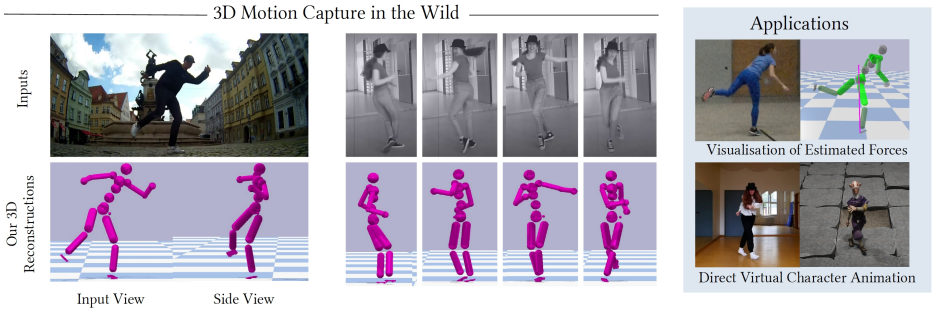


Figure 4.1: (Left:) Results of our method on different sequences from the input and side views. (Right:) Applications in motion analysis by force visualisation and virtual character animation.

uration and scale from 2D cues, as well as due to difficult (self-)occlusions (Kocabas et al., 2020a; Martinez et al., 2017; Mehta et al., 2017b; Pavlakos et al., 2018b). Most state-of-the-art methods for 3D human pose estimation and motion capture benefit from the rapid progress in machine learning and have shown stark improvements in accuracy (Kocabas et al., 2020a; Sun et al., 2019; Wandt and Rosenhahn, 2019). Despite this progress, existing purely kinematic methods still have important limitations and produce notable artefacts. Many produce per-frame predictions that can be temporally highly unstable, and many produce root-relative but not global 3D poses. Further, most existing methods are, by design, incapable of considering interactions with the environment, let alone biophysical pose or motion plausibility. The former often leads to collision violations such as foot-floor penetration and floating in the air in captured motions; the latter yields impossible poses with physically incorrect leaning and posture or poses that would actually cause loss of balance. Captured results are, therefore, not only inaccurate in several ways but also unnatural, which greatly reduces data usability, in particular in computer graphics related applications.

We, therefore, propose a new neural network-based approach for monocular 3D human motion capture, which considers physical constraints in the observed scenes. We believe that improving upon the recently proposed ideas of physical-awareness constraints in monocular 3D human motion capture (Rempe et al., 2020; Shimada et al., 2020) and combining them with machine learning techniques can lead to further advances in the domain. While the methods of Rempe et al. (2020) and Shimada et al. (2020) contain two stages—with the physics-based pose

optimisation implemented as an engineered method relying on classical optimisation techniques,—*we are the first to propose a fully-differentiable framework for monocular 3D human motion capture with physical awareness*. Thus, our physics-based pose optimisation is a trainable neural network with custom layers for physics-based constraints. We refer to our approach as *physionical*, which means that it is fully differentiable, neural network-based and aware of physical boundary conditions. The 3D motions estimated by our framework are smooth and natural and can directly drive an animation character with no further postprocessing. We can also visualise the joint torques and ground reaction forces estimated from the motion in the video, which can be used for some applications, *e.g.* sports analysis. See Fig. 4.1 for the visualisation of the reconstructed 3D motions and the example applications of our framework.

Our method includes two core neural components, *i.e.* a *target pose estimator network* (TPNet) and an iterative *dynamic cycle* for controlling a humanoid character while considering physics-based boundary conditions. Both TPNet and the dynamic cycle are newly developed neural networks that are end-to-end trained. TPNet kinematically regresses the target reference 3D pose from input 2D keypoints that are obtained by an off-the-shelf 2D detector, which serves as a foundation for the dynamic cycle. The dynamic cycle first calculates the gain parameters of a neural proportional-derivative (PD) controller, generating a force vector to control the kinematic character with physics properties through the differentiable physics model. The force vector is then used to estimate the ground reaction force (GRF), and both are then passed to the forward dynamics module, which regresses the accelerations of the skeleton. The latter is subsequently used to update the final global human pose in 3D, which matches the subject's 2D pose in the input frames and obeys the condition of plausible foot-floor placements. In the dynamic cycle, our architecture contains a novel differentiable layer realising a *hard* constraint for preventing foot-floor penetration. Our motivation for a custom optimisation layer comes from the fact that conventional losses in neural networks can only express soft constraints on the learned manifold, *i.e.* there is no guarantee that the expressed boundary conditions will be strictly fulfilled at inference time. On the other hand, physical constraints and forces such as gravity and ground reaction force (originating from

the floor which naturally limits human motions) are strictly present in the physical world without freedom of interpretation.

Since our architecture is fully differentiable, it is the first approach for monocular physics-aware 3D motion capture that can be equally trained on images annotated with strong and weak labels, *i.e.* joint angles, 3D joint keypoints but also 2D joint keypoints. Since also 2D training data can be used, our method can be trained for better generalisation and is easier to fine-tune for motion classes for which any 3D annotation would be very hard (*e.g.* in-the-wild athletics or sports videos). Our physionical method is aware of the environment and physical laws and runs in real-time at 20 frames per second. It outputs physically plausible results with significantly fewer artefacts—such as unnatural temporal instabilities and frame-to-frame jitter, foot-floor penetration and the uncertainty in the absolute human poses along the depth dimension—than purely kinematic methods and other physics-aware methods. Moreover, compared to the most related method PhysCap (Shimada et al., 2020) in the previous chapter, we mitigate the delay between the observed and estimated motions. To summarise, the technical **contributions** of this chapter are as follows:

- The first entirely neural and fully-differentiable approach for markerless 3D human motion capture from monocular videos with physics constraints, which we call physionical (Sec. 4.3).
- A new canonicalisation of input 2D keypoints allowing network training and 3D human pose regression with different intrinsic camera parameters and jointly on several datasets (Sec. 4.3.2). In contrast to existing normalisation methods for human pose estimation in the literature, our canonicalisation does not discard the cues for the global pose estimation.
- The integration of hard boundary conditions in our architecture to prevent foot-floor penetrations by taking advantage of the recent progress in designing optimisation layers for neural architectures (Agrawal et al., 2019a) (Sec. 4.3.4).
- Applications of our method in direct virtual character animation and visualisation of joint torques related to muscle activation forces,

which can be used to analyse the captured motions in conceivable downstream tasks (Sec. 4.3.7).

The proposed method establishes a new state of the art and outperforms existing methods on several metrics, as shown in the experiments (Sec. 4.5). We evaluate it on several datasets including Human3.6M (Ionescu et al., 2013), MPI-INF-3DHP (Mehta et al., 2017a), DeepCap (Habermann et al., 2020) as well as newly-recorded sequences (Sec. 4.5). The differences in the results of our physionical approach compared to existing techniques are especially noticeable when they are obtained on scenes in the wild.

4.2 RELATED WORK

A vast body of literature is devoted to 3D human motion capture with multi-view systems (Bo and Sminchisescu, 2008; Brox et al., 2010; Elhayek et al., 2015; Gall et al., 2010; Martin-Brualla et al., 2018; Starck and Hilton, 2007; Wu et al., 2012) and inertial on-body sensors (Dejnabadi et al., 2006; Marcard et al., 2017; Tautges et al., 2011; Vlasic et al., 2007). Both areas are well studied and these methods have shown impressive results. On the downside, they require specialised camera rigs and hardware which make their operation outside the studio difficult. In this section, we thus further focus on related works on 1) physics-based virtual character animation and 2) monocular 3D human pose estimation and motion capture.

4.2.1 *Physics-Based Virtual Character Animation*

Many works have been proposed for physics-based character animation which is a significantly different problem compared to monocular 3D human motion capture. In virtual character animation, there is full control over the simulated physical laws and the structure of the simulated world (in which virtual characters are moving), whereas we are interested in reconstructing physically plausible human motions from partial observations (monocular videos). At the same time, the animated character of these methods is inspirational for us, as they provide the realism and motion plausibility of character motion required in computer

graphics applications (Andrews et al., 2016; Barzel et al., 1996; Bergamin et al., 2019; Levine and Popović, 2012; Liu et al., 2010; Sharon and Panne, 2005; Wrotek et al., 2006; Zheng and Yamane, 2013). Some techniques for virtual character animation employ deep reinforcement learning and motion imitation in physics engines, often requiring specialised networks for each motion kind (Bergamin et al., 2019; Jiang et al., 2019; Lee et al., 2019; Peng et al., 2018a,b). In contrast to the latter, our problem requires a different approach. Since our goals are the generalisability across different motions and high data throughout enabling real-time applications, we use explicit equations of motions and physics-based constraints on top of initial kinematic estimates, while preserving the differentiability of our architecture trained in a supervised manner.

4.2.2 *Classical Monocular 3D Human Motion Capture and Pose Estimation*

This section focuses on the majority of works on monocular 3D human motion capture and pose estimation that do not use explicit physics-based and environment constraints. All such methods for 3D human pose estimation and motion capture can be classified into 1) direct regression approaches, 2) lifting approaches and 3) various hybrid approaches leveraging mixtures of 3D and 2D predictions. The first category of methods is based on convolutional neural networks and regresses 3D joints directly from input images (Mehta et al., 2017a; Rhodin et al., 2018; Tekin et al., 2016). The methods of the second category regress 3D joints from detected 2D keypoints (Chen and Ramanan, 2017; Martinez et al., 2017; Moreno-Noguer, 2017; Pavlakos et al., 2018b; Tomè et al., 2017). Finally, multiple methods combine 3D joint depth (or location probabilities) and 2D keypoint prediction with lifting constraints (Habibie et al., 2019; Mehta et al., 2017b; Newell et al., 2016; Pavlakos et al., 2017; Yang et al., 2018; Zhou et al., 2017). Among them, Habibie et al. (2019) use additional weak supervision with in-the-wild images.

Some methods additionally use 3D shape priors. Statistical human body models provide strong constraints on plausible human postures which can be used for human pose estimation (Bogo et al., 2016; Kanazawa et al., 2018; Kocabas et al., 2020a). In contrast, certain works leverage actor-specific 3D human body templates for global 3D human motion capture with shape tracking including surface deformations on top of

a skeletal motion (Habermann et al., 2020; Xu et al., 2020; Xu et al., 2018). Several further algorithms use different variants of anatomical constraints for the human body (*e.g.* body symmetry) and show improved results in weakly-supervised (Dabral et al., 2018; Wandt and Rosenhahn, 2019) or even unsupervised 3D human pose estimation (Kovalenko et al., 2019). Some works also use geometric vicinity and collision avoidance constraints for the reconstruction of human-object interactions (Hassan et al., 2019; Zhang et al., 2020a), and several other works can generalise to multiple persons in the scene (Dabral et al., 2019; Fabbri et al., 2020; Mehta et al., 2020; Rogez et al., 2019; Zanfir et al., 2018).

Most of the proposed algorithms work on single images (Kanazawa et al., 2018; Kolotouros et al., 2019; Pavlakos et al., 2018b; Song et al., 2020; Sun et al., 2019), whereas others take the temporal information into account for improved temporal stability (Kanazawa et al., 2019; Kocabas et al., 2020b; Pavllo et al., 2019). To directly drive a kinematic character with skinned rigs, we need joint angles, root translation and rotation of a consistent skeleton. Only few works estimate those from the input RGB video directly and realise the character motion control from a video (Mehta et al., 2020, 2017b; Shi et al., 2020). Among the latter, MotioNet (Shi et al., 2020) is the most closely related method to ours. Unlike our approach, it does not include an explicit physics model, which can adds up to physically implausible effects in the estimates. Upon the architecture design, MotioNet expects, at test time, the same intrinsic camera parameters as in the training dataset, *i.e.* when the system is applied to sequences with different camera intrinsics, the accuracy of the estimated translations can vary considerably. In contrast, we use canonical 2D keypoints which makes our physionical approach invariant to camera intrinsics.

4.2.3 *Monocular 3D Human Motion Capture with Physics-based Constraints*

This section focuses on the emerging field of monocular 3D human motion capture with physics-based constraints. One of the pioneering works in this domain was proposed by Wei and Chai (2010). Their method requires manual user interactions for each input sequence and is computationally expensive. Vondrak et al. (2012) perform 3D human motion capture from monocular videos for physically-plausible character control.

They recover 3D bipedal controllers using optimal control theory, which are capable of simulating the observed motions in different environments. Unfortunately, this method cannot easily generalise across motions and does not run in real time. Zell et al. (2017) estimate 3D human poses along with the inner and exterior forces from images for object lifting and walking. Li et al. (2019) regress human and object poses in 3D along with forces and torques exerted by human limbs from a monocular video and an object prior. They focus on instruments with grips and recognise contacts between a person and an object (*i.e.* the instrument or the ground) to facilitate the trajectory-optimisation problem. The method of Zell et al. (2020) for the analysis of 3D human motion capture relates to our setting. It infers ground-reaction forces and joint torques from input 3D human motion capture sequences, relying on a new dataset with multiple human motion types and ground-truth forces acquired using force plates on the floor. The advantage of this method is that the proposed forward and inverse dynamics layers generalise to new locomotion types. Thus, the main focus lies on the *explainability* of the captured human motions in 3D from the physics perspective, whereas our goal is 3D human motion capture that satisfies physics-based (environmental) constraints at interactive framerates.

Two recent methods for monocular 3D human motion capture with physics constraints are Rempe et al. (2020) and Shimada et al. (2020). They tackle general human motions by introducing laws of physics as regularisers in their formulations. Both methods 1) start with initial kinematic estimates (Xiang et al. (2019) and Mehta et al. (2017b), respectively) which are subsequently refined through physics-based optimisation, 2) detect foot contacts and 3) the camera is not moving. Rempe et al. (2020) and Shimada et al. (2020), however, differ significantly in physics-based global pose optimisation and the overall runtime. Rempe et al. (2020) use as a proxy a reduced-dimensional model of the lower body inspired by Winkler et al. (2018), which does not include all joints but captures the overall motion and contacts. In contrast, Shimada et al. (2020) rely on initial kinematic pose corrections and a lightweight iterative physics-based pose refinement with PD joint controllers and ground-reaction force estimation, which enable real-time operation. Both these approaches are compositional and only partially rely on neural networks (for the kinematic estimates and foot contact detections, but not for the physics-based rea-

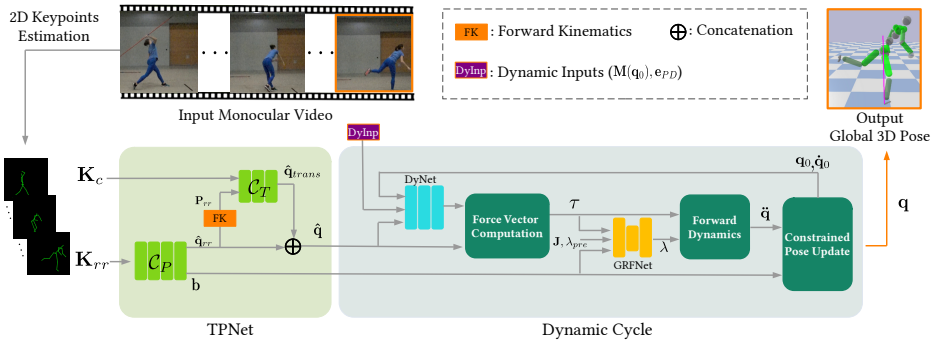


Figure 4.2: Overview of our physionical approach for markerless monocular 3D human motion capture.

soning), unlike our approach. We embed hard physics-based constraints in our architecture and enable its full differentiability. Our trainable model with explicit physics-based constraints realises more plausible 3D motion qualitatively and more accurate 3D poses quantitatively than the existing physics-based approaches solving conventional optimisation problems with the dynamics equations of motion (see Sec. 4.5).

4.3 METHOD

OVERVIEW Our goal is physically plausible monocular global 3D human motion capture without markers. We follow a learning-based approach trained through a fully-differentiable physics model, see Fig. 4.2 for an overview. Our framework includes a neural proportional-derivative (PD) controller that estimates a force vector, allowing controlling the kinematic character with dynamics properties to match its pose with the subject's pose in the image sequence. The ground reaction forces are also estimated alongside the 3D motions without requiring any supervisory force annotations. We can also read out and visualise internal and contact forces regressed from the monocular input. Our method accepts sequential 2D joint keypoints in a video (*e.g.* extracted with an of-the-shelf 2D keypoint detector), and returns 3D skeleton poses that satisfy (bio-)physical constraints. This significantly mitigates foot-floor penetration, body sliding along depth direction and joint jitters. In Sec. 4.3.1, we define our model and mathematical notations. In Sec. 4.3.2, we discuss a canonicalisation method of the input 2D joint keypoints which allows our global translation estimation network $\mathcal{C}_T$ to be trained

jointly on several datasets with different camera intrinsics. In Secs. 4.3.3 and 4.3.4, the target pose estimation network and the dynamic cycle with physics-based constraints are elaborated, respectively. In the latter, the 3D pose is updated in the custom optimisation layer where we introduce a hard constraint to prevent foot-floor penetration in a differentiable manner. The obtained 3D poses are smooth, plausible and show mitigated motion delay even on fast motion sequences thanks to the learning-based PD controller which dynamically adjusts the gain parameters depending on the motions in the scene. Our fully-differentiable architecture allows finetuning using 2D annotations only for improved accuracy on in-the-wild footage (Sec. 4.3.6). Applications of our methods are discussed in Sec. 4.3.7.

4.3.1 *Our Model, Assumptions and Notations*

We represent the kinematic state of the skeleton by a pose vector $\mathbf{q} \in \mathbb{R}^{n+1}$ and its velocity $\dot{\mathbf{q}} \in \mathbb{R}^n$ in the camera frame, with $n = 46$. The first seven entries of $\mathbf{q}$ represent the root translation $\mathbf{q}_{\text{trans}} \in \mathbb{R}^3$ and rotation in the quaternion parametrisation $\mathbf{q}_{\text{ori}} \in \mathbb{R}^4$, respectively. All remaining $n-7$ entries of $\mathbf{q}$ encode joint angles of the human skeleton model parametrised by Euler angles. The first three entries of $\dot{\mathbf{q}}$ represent the linear velocity of the root whereas the next three ones stand for its angular velocity $\omega \in \mathbb{R}^3$. The remaining entries of $\dot{\mathbf{q}}$ stand for the angular velocity of each joint and they correspond to the joint order in $\mathbf{q}$. The time derivative of $\mathbf{q}_{\text{ori}}$ is approximated as follows:

$$\frac{\mathrm{d}\mathbf{q}_{\text{ori}}}{\mathrm{d}\mathbf{t}} \approx \frac{1}{2} \begin{bmatrix} 0 \\ \omega \end{bmatrix} \otimes \mathbf{q}_{\text{ori}}, \tag{4.1}$$

where $\otimes$ represents a quaternion multiplication. Eq. (4.1) is used to update the 3D root orientation from its angular velocity in each dynamics simulation step.

We use M 2D joint keypoints normalised in two different ways, *i.e.* the root-relative 2D keypoints normalised by the image size and gathered in $\mathbf{K}_{rr} \in \mathbb{R}^{M\times 2}$, and the canonical 2D keypoints stacked in $\mathbf{K}_c \in \mathbb{R}^{M\times 2}$, allowing the network training on datasets with different camera intrinsics. Resorting to root-relative 2D joint keypoints is a widely-used normali-

sation approach for estimating the root-relative 3D pose from an image or video since it is translation-invariant in the image space. Therefore, we use $\mathbf{K}_{rr}$ for estimating the joint angles and root orientation of the character $\mathbf{q}_{rr}$. While this normalisation alone loses the cues for estimating the global translation of the subject in the scene, the canonicalised 2D joint keypoints retain the required information to regress the global pose, see Sec. 4.3.2 for the details.

Our character is composed of *links* which are volumetric body part representations with collision proxies, following the same structure as Shimada et al. (2020). Our core idea is to enable awareness of physical laws in our framework which helps to obtain physically plausible human motion captures. We impose the laws of physics by considering Newtonian rigid body dynamics, which—when applied to our case—reads as (Featherstone, 2014):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \boldsymbol{\tau} = \mathbf{J}^{\mathsf{T}}(\mathbf{q})\mathbf{G}\boldsymbol{\lambda} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), \tag{4.2}$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\ddot{\mathbf{q}} \in \mathbb{R}^n$ are the inertia matrix in the skeleton frame, which describes the moments of inertia of the system, and the acceleration of $\mathbf{q}$, respectively; $\mathbf{J} \in \mathbb{R}^{6N_c \times n}$ is a contact Jacobian matrix which relates velocities in the skeleton frame to velocities in Cartesian coordinates; N_c denotes the number of links to which the contact forces are applied; $\mathbf{G} \in \mathbb{R}^{6N_c \times 3N_c}$ is the matrix that converts the contact force $\boldsymbol{\lambda} \in \mathbb{R}^{3N_c}$ to linear forces and torques (for its details, readers are referred to Featherstone (2014)); $\mathbf{h} \in \mathbb{R}^n$ encompasses gravity, Coriolis and centripetal forces; the force vector $\boldsymbol{\tau} \in \mathbb{R}^n$ represents the internal joint forces of the character, with its first six entries being the direct root actuations which are set to 0 as per convention.

The total forces that explain the root motion include external forces such as ground reaction force (GRF). Similar to several prior works (Andrews et al., 2016; Levine and Popović, 2012; Shimada et al., 2020; Yuan and Kitani, 2020), we minimise the direct (virtual) root actuation by estimating the acting GRF and explaining the observed motions with it as much as possible (instead of setting the first six entries of $\boldsymbol{\tau}$ to zero).

4.3.2 *Input Canonicalisation*

For the networks that estimate the character's pose without global translation $\mathbf{q}_{rr}$ (*e.g.* C_P), we use root-relative 2D joint keypoints $\mathbf{K}_{rr}$. Many algorithms, which use a perspective camera model, estimate the global root position by optimising a 2D projection-based loss without learning components (Habermann et al., 2020; Mehta et al., 2020, 2017b). Pavllo et al. (2019) and Shi et al. (2020) employ neural networks to directly regress the translation of the 3D poses. However, in this case, the learned motion manifolds depend on the camera intrinsic parameters used during the training. Consequently, at test time, they expect similar camera intrinsics. To tackle this issue, we propose to use canonicalised 2D keypoints $\mathbf{K}_c$ to factor out the influence of the camera intrinsics before they are fed to the neural network that regresses the absolute root translation of the character. Our architecture benefits from the canonicalisation in two ways. First, the translation estimation network can be trained with a large scale joint dataset, *i.e.* a composition of Human 3.6M (Ionescu et al., 2013), MPI-INF-3D-HP (Mehta et al., 2017a) and DeepCap (Habermann et al., 2020), which are recorded with different intrinsic camera parameters. Second, arbitrary camera intrinsics can be used at test time without influencing the performance of the network that regresses the global translations.

Consider the perspective camera projection without a skew parameter:

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{f_x X}{Z} + c_x \\ \frac{f_y Y}{Z} + c_y \\ 1 \end{bmatrix}, \tag{4.3}$$

where $[X, Y, Z]^\mathsf{T}$ is a 3D coordinate of a joint in the camera frame, f the focal length and c the principal point. We see that the 2D joint keypoints $\left[\frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y\right]^\mathsf{T}$ are influenced by the camera intrinsic parameters. Therefore, we generate canonical 2D joint keypoints by applying the identity as an intrinsic camera matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix}. \tag{4.4}$$

We use $[X/Z, Y/Z]^\mathsf{T}$ as the canonical 2D joint keypoint which is not influenced by camera intrinsic parameters. In the case, when the depth information Z is not known (*e.g.* during the testing phase), we can still obtain the canonical 2D joint keypoints assuming that camera intrinsics are known. Let $p_m = [u_m, v_m]^\mathsf{T}$ be the 2D joint locations of M joints in the pixel coordinates, with $m \in \{1, \ldots, M\}$. We next stack the canonicalised 2D keypoints in a single $\mathbf{K}_c$ matrix:

$$\mathbf{K}_c = \begin{bmatrix} \frac{u_1 - c_x}{f_x} & \frac{u_2 - c_x}{f_x} & \ldots & \frac{u_M - c_x}{f_x} \\ \frac{v_1 - c_y}{f_y} & \frac{v_2 - c_y}{f_y} & & \frac{v_M - c_y}{f_y} \end{bmatrix}^\mathsf{T}. \tag{4.5}$$

It follows from Eqs. (4.4) and (4.5), that for a single p_m and for the corresponding 3D joint location $P_m = [X_m, Y_m, Z_m]^\mathsf{T}$, we have:

$$\left[\frac{u_m - c_x}{f_x}, \frac{v_m - c_y}{f_y}\right]^\mathsf{T} = \left[\frac{X_m}{Z_m}, \frac{Y_m}{Z_m}\right]^\mathsf{T}. \tag{4.6}$$

This can be interpreted as a point lying on the plane with $Z = 1$. The generalisability and accuracy of the networks trained with the canonicalised 2D keypoints are evaluated in Sec.4.5.

4.3.3 *Target Pose Estimation*

After pre-processing the 2D joint keypoints (Sec. 4.3.2), the inputs are fed to the target pose estimation network (TPNet) that outputs the global target pose $\hat{\mathbf{q}} \in \mathbb{R}^{n+1}$ and binary labels for the contact states, *i.e.* toes and heels $\mathbf{b} \in \{0, 1\}^4$, see Fig. 4.2 for the overview. TPNet consists of two 1D convolution-based network modules ($\mathcal{C}_T$ and $\mathcal{C}_P$) that consider temporal information. Network $\mathcal{C}_P$ first estimates the joint angles and global orientation of the character without the root translation, which is denoted by $\hat{\mathbf{q}}_{rr}$, and foot contact labels $\mathbf{b}$ in the scene; $\hat{\mathbf{q}}_{rr}$ is further processed by the forward kinematics layer $f(\cdot)$ to obtain the root-relative 3D joint keypoints $\mathbf{P}_{rr}$ with bone lengths in Cartesian coordinates in the absolute scale. Network $\mathcal{C}_T$ takes as input $\mathbf{P}_{rr}$ and $\mathbf{K}_c$, and outputs the global translation of the character $\hat{\mathbf{q}}_{\text{trans}}$. At the end, we obtain global 3D skeleton pose $\hat{\mathbf{q}}$ which is further employed as a target pose of the PD controller (Sec. 4.3.4.1). All the networks in TPNet are composed of four residual blocks with 1D convolution layers with a window size of

10. Note that our networks accept only past and current frames with no access to the future frames, hence compatible with real-time applications.

4.3.4 *Dynamic Cycle*

In this section, we elaborate the dynamic cycle of our framework where we control the human character considering dynamics quantities: $\mathbf{M}$, $\mathbf{J}$ and $\mathbf{h}$ are analytically estimated in each simulation step using the current pose $\mathbf{q}_0$ and the velocity $\dot{\mathbf{q}}_0$ (Featherstone, 2014).

4.3.4.1 *Force Vector Computation by a Neural PD Controller*

PD controllers enable motion tracking with a kinematic character while maintaining a smooth motion. They are hence widely used in robotics and physics-based animation research (Chentanez et al., 2018; Lee et al., 2019; Levine and Popović, 2012; Putri, Machbub, et al., 2018; Sugihara and Nakamura, 2006). Our framework also utilises a PD controller to compute the internal force vector $\boldsymbol{\tau}$ of the character. However, the smoothing properties of PD controller can cause motion delay in the presence of fast motions if the gain values are not optimal. The motion delay is especially apparent when the results are shown reprojected to the input views. This issue arises from fixing the gains which adjust the PD controller's sensitivity to the pose and velocity error (Shimada et al., 2020).

Similarly to Chentanez et al. (2018), we dynamically change the gain coefficients depending on the target and current skeleton poses by our dynamics network (DyNet). This approach significantly mitigates the motion delay compared to the existing methods while keeping the motions smooth. Our DyNet accepts the target pose $\hat{\mathbf{q}}$, the current pose $\mathbf{q}_0$, the current velocity $\dot{\mathbf{q}}_0$, the mass matrix $\mathbf{M}$ and the current pose error $\mathbf{e}_{\text{PD}} = d(\hat{\mathbf{q}}, \mathbf{q}_0) \in \mathbb{R}^n$, and outputs gain parameters $\mathbf{k}_p \in \mathbb{R}^n$ of the PD controller along with the offset forces $\alpha \in \mathbb{R}^n$ for each DoF. The error function $d(\cdot)$ computes entry-wise differences between $\hat{\mathbf{q}}$ and $\mathbf{q}_0$. For the entries that represent the root orientation, we compute the quaternion difference. Since we provide $\hat{\mathbf{q}}$ and $\mathbf{q}_0$, their residual information, *i.e.* $\mathbf{e}_{\text{PD}}$, is not the essential input for the network. However, similar to Bergamin et al. (2019), we observed that explicitly providing the current error to DyNet leads to a much faster loss convergence. Therefore, we include

$\mathbf{e}_{\text{PD}}$ as one of the inputs to DyNet. The outputs of TPNet and DyNet are used to compute the force vector $\boldsymbol{\tau}$ following the PD controller rule with the compensation term $\mathbf{h}$[1].:

$$\boldsymbol{\tau} = \mathbf{k}_p \circ (\hat{\mathbf{q}} - \mathbf{q}_0) - \mathbf{k}_d \circ \dot{\mathbf{q}}_0 + \alpha + \mathbf{h}, \tag{4.7}$$

where "$\circ$" denotes Hadamard matrix product. $\mathbf{h}$ represents the sum of gravity, centripetal and Coriolis forces, which are analytically computed.

4.3.4.2 *Ground Reaction Force Estimation*

In the real world, external forces are required to control the centre of gravity of a human body. In other words, for the motion to be physical, the global translation and rotation of the character need to be controlled by external forces such as ground reaction forces obtained from the contact positions. On the other hand, the character motion can be controlled to match the pose of the subject in the scene using the force vector $\boldsymbol{\tau}$. However, $\boldsymbol{\tau}$ contains direct linear and rotational force applied on the root position as elaborated in Sec. 4.3.4.1.

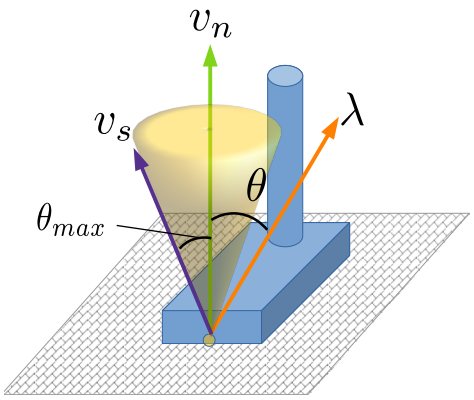


Figure 4.3: Schematic visualisation of the friction cone and the ground reaction force (GRF) at the foot-floor contact position.

We thus train the ground reaction force estimation network (GRFNet) to minimise the (virtual) force applied directly on the root, trying to explain the global motion by the ground reaction force $\boldsymbol{\lambda}$ as much as possible. Let $\boldsymbol{\tau}_{\text{root}} \in \mathbb{R}^6$ be the force vector corresponding to the root position (*i.e.* the first six elements of $\boldsymbol{\tau}$).

Then, the main objective function for training GRFNet reads:

$$\mathcal{L}_{\text{force}} = \left\| \boldsymbol{\tau}_{\text{root}} - \mathbf{J}_1^{\mathsf{T}} \mathbf{G} \boldsymbol{\lambda} \right\|_2^2, \tag{4.8}$$

1 In literature, this is known as PD controller with force compensation (Yang et al., 2010).

where $\mathbf{J}_1^\top$ denotes the first six rows of $\mathbf{J}^\top$ corresponding to the root configuration. Minimising Eq. (4.8) encourages the network to estimate $\boldsymbol{\lambda}$ which explains the forces applied on the root position by GRF.

However, the direction of the contact force does not only depend on Eq. (4.8). Therefore, we also introduce the friction constraint $\mathcal{F}$ for $\boldsymbol{\lambda}$ to be physically plausible. The estimated $\boldsymbol{\lambda}$ needs to be inside of the so-called friction cone which is defined by the friction coefficient $\mu = 0.8$ together with the normal and tangential directions of the contact position. The friction-cone constraint is defined as follows:

$$\mathcal{F}^\ell = \left\{ \boldsymbol{\lambda}^\ell \in \mathbb{R}^3 \;\middle|\; \lambda_n^\ell > 0, \left\| \lambda_t^\ell \right\|_2 \leq \mu \lambda_n^\ell \right\}, \tag{4.9}$$

where ℓ represents the identifier of the link where contact force is applied; λ_n and λ_t represent the normal and tangential component of λ, respectively. We next reformulate Eq. (4.9) to integrate into the training objective of GRFNet:

$$\mathcal{L}_{\text{cone}} = \begin{cases} \|\theta\|_2^2, & \text{if } \theta > \theta_{\max}, \\ 0, & \text{else}, \end{cases} \tag{4.10}$$

where $\theta_{\max}$ is the angle between the normal vector v_n of the contact position and a vector v_s that lies on the surface of the friction cone, and θ is the angle between v_n and $\boldsymbol{\lambda}$, see Fig. 4.3. Next, we introduce a temporal smoothness regulariser for the ground reaction force $\boldsymbol{\lambda}$:

$$\mathcal{L}_{\text{smooth}} = \left\| \boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{pre}} \right\|_2^2, \tag{4.11}$$

where $\boldsymbol{\lambda}_{\text{pre}}$ represents the estimated $\boldsymbol{\lambda}$ in the previous simulation step. The final objective function for GRFNet $\mathcal{L}_{\text{GRF}}$ is as follows:

$$\mathcal{L}_{\text{GRF}} = \mathcal{L}_{\text{force}} + \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{cone}}. \tag{4.12}$$

4.3.4.3 *Forward Dynamics*

To introduce the laws of physics in our 3D motion capture algorithm, we embed the forward dynamics layer in our architecture. We derive joint accelerations $\ddot{\mathbf{q}}$ from Eq. (4.2):

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})(\boldsymbol{\tau}^* + \mathbf{J}^\mathsf{T}\mathbf{G}\boldsymbol{\lambda} - \mathbf{h}), \tag{4.13}$$

where $\boldsymbol{\tau}^* = \boldsymbol{\tau} - \mathbf{J}^\mathsf{T}\mathbf{G}\boldsymbol{\lambda}$. In this formulation, $\boldsymbol{\tau}^*$ expresses the minimised direct root actuation with contact force compensation for each joint torque. This forward dynamics layer returns $\ddot{\mathbf{q}}$ considering the mass matrix of the body $\mathbf{M}$, internal and external forces, gravity, Coriolis and centripetal forces encompassed in $\mathbf{h}$.

4.3.4.4 *Constrained Pose Update*

In this step, we update the character's pose using the estimated accelerations $\ddot{\mathbf{q}}$ through the differentiable optimisation layer to prevent foot-floor penetration. Given $\ddot{\mathbf{q}}$ in the skeleton frame and the simulation time step Δt, the velocity $\dot{\mathbf{q}}$ and the kinematic 3D pose $\mathbf{q}$ are updated using the finite differences:

$$\begin{aligned} \dot{\mathbf{q}}^{i+1} &= \dot{\mathbf{q}}^{i} + \Delta t\, \ddot{\mathbf{q}}^{i}, \\ \mathbf{q}^{i+1} &= \mathbf{q}^{i} + \Delta t\, \dot{\mathbf{q}}^{i+1}, \end{aligned} \tag{4.14}$$

where i denotes the simulation step identifier. To prevent foot-floor penetration, we introduce the differentiable optimisation layer following the formulation of Agrawal et al. (2019b). This custom neural network layer solves a specific optimisation problem for each forward pass and returns its derivatives for each backward pass. More specifically, we update the velocity in the skeleton frame $\dot{\mathbf{q}}$ solving the optimisation below:

$$\min_{\dot{q}^*} \left\| \dot{\mathbf{q}}^* - \dot{\mathbf{q}} \right\|, \text{ s.t. } \mathbf{r}_n^c > 0, \tag{4.15}$$

where $\mathbf{r}_n^c$ represents the linear velocity of the contact position along the normal direction of the floor. Velocity vector $\mathbf{r}^c$ is computed as follows:

$$\mathbf{r}^c = \mathbf{T}(\mathbf{J}\dot{\mathbf{q}}), \tag{4.16}$$

where $\mathbf{T}(\cdot)$ is the transformation from the camera frame to the floor frame of reference. After solving (4.15), the estimated $\dot{\mathbf{q}}^*$ is substituted as $\dot{\mathbf{q}}$ in

Eq. (4.14). The dynamic cycle introduced in this section is iterated $k = 6$ times. After the iterations are complete, we obtain the final physically plausible 3D character's pose $\mathbf{q}$.

4.3.5 *Network Training*

We pre-train TPNet for a more stable training of the whole architecture. Such pre-training is advantageous due to two reasons. First, estimating joint angles from 2D joint keypoints leads to ambiguities in bone orientations (Shi et al., 2020). Second, controlling the dynamic character in 3D by estimated forces to match the subject's pose only from 2D joint keypoints is an ill-posed problem. The network $\mathcal{C}_P$ in TPNet is pre-trained with the following objective loss function:

$$\mathcal{L}_{\mathcal{C}_P} = \mathcal{L}_{3D}(\hat{\mathbf{q}}) + \mathcal{L}_{2D}(\hat{\mathbf{q}}) + \mathcal{L}_{\text{ori}}(\hat{\mathbf{q}}_{\text{ori}}) + \mathcal{L}_{\text{irr.}}(\hat{\mathbf{q}}) + \mathcal{L}_b(\mathbf{b}). \tag{4.17}$$

The main 3D loss $\mathcal{L}_{3D}$ is defined as follows:

$$\mathcal{L}_{3D}(\hat{\mathbf{q}}) = \left\| f(\hat{\mathbf{q}}) - \mathbf{p}'_{3D} \right\|_2^2, \tag{4.18}$$

where $f(\cdot)$ and $\mathbf{p}'_{3D}$ are forward kinematics function and ground-truth 3D joint keypoints, respectively. The loss $\mathcal{L}_{2D}$ stands for the 2D reprojection error:

$$\mathcal{L}_{2D}(\hat{\mathbf{q}}) = \left\| \Pi(f(\hat{\mathbf{q}})) - \mathbf{p}'_{2D} \right\|_2^2, \tag{4.19}$$

where $\Pi(\cdot)$ and $\mathbf{p}'_{2D}$ are the perspective projection operator and ground-truth 2D joint keypoints normalised by the image size, respectively. The loss $\mathcal{L}_{\text{ori}}$ is added for the supervision of the global root orientation represented by a quaternion:

$$\mathcal{L}_{\text{ori}}(\hat{\mathbf{q}}_{\text{ori}}) = \left\| \hat{\mathbf{q}}_{\text{ori}} \ominus \mathbf{q}'_{\text{ori}} \right\|_2^2, \tag{4.20}$$

where $\mathbf{q}'_{\text{ori}}$ is the ground-truth root orientation in quaternion parametrisation, and "$\ominus$" denotes a difference computation after converting the quaternion into a rotation matrix. The loss $\mathcal{L}_{\text{irr.}}$ keeps the estimated joint angles in a reasonable range:

$$\mathcal{L}_{\text{irr.}}(\hat{\mathbf{q}}) = \sum_{i=1}^{40} \Psi(\hat{\mathbf{q}}_i), \text{ with} \tag{4.21}$$

$$\Psi(\hat{\mathbf{q}}_i) = \begin{cases} (\hat{\mathbf{q}}_i - \psi_{\max,i})^2, & \text{if } \hat{\mathbf{q}}_i > \psi_{\max,i} \\ (\psi_{\min,i} - \hat{\mathbf{q}}_i)^2\,, & \text{if } \hat{\mathbf{q}}_i < \psi_{\min,i}, \\ 0 \qquad\qquad\quad , & \text{otherwise,} \end{cases} \tag{4.22}$$

where $\hat{\mathbf{q}}_i$ denotes the joint angle of the i-th joint and $[\psi_{\min,i}, \psi_{\max,i}]$ defines the reasonable angle range for the i-th joint. Term $\mathcal{L}_b$ is the binary cross entropy loss to train the network for estimating correct foot contact states in the scene:

$$\mathcal{L}_b(\mathbf{b}) = -\sum_{i=1}^{4} b'_i \log(b_i) + (1 - b'_i) \log(1 - b_i), \tag{4.23}$$

where b'_i and b_i are the ground-truth contact label and predicted contact probability on i-th joint, respectively.

The $\mathcal{C}_T$ module of TPNet is trained with the 3D translation loss:

$$\mathcal{L}_{C_T}(\hat{\mathbf{q}}_{\text{trans}}) = \left\| \hat{\mathbf{q}}_{\text{trans}} - \mathbf{q}'_{\text{trans}} \right\|_2^2, \tag{4.24}$$

where $\mathbf{q}'_{\text{trans}}$ denotes the ground-truth translation in 3D space.

After pre-training $\mathcal{C}_P$ and $\mathcal{C}_T$ with $\mathcal{L}_{C_T}$ and $\mathcal{L}_{C_P}$, we train DyNet with the following loss:

$$\mathcal{L}_{\text{Dyn}}(\mathbf{q}) = \|\mathbf{q} - \hat{\mathbf{q}}\|_2^2 + \varphi \|\boldsymbol{\tau}\|_2^2, \tag{4.25}$$

where $\mathbf{q}$ is the final, physically-plausible 3D pose passed through the differentiable physics model and $\varphi = 10^{-6}$ is the weight of the regularisation term of $\boldsymbol{\tau}$. The first term of $\mathcal{L}_{\text{Dyn}}$ enforces the character to catch up with the target pose with the mitigated motion delay by dynamically estimating the gain parameters of the PD controller. The second term of $\mathcal{L}_{\text{Dyn}}$ prevents overshooting of the PD controller output. The GRFNet is trained with Eq. (4.12) (Sec. 4.3.4.2). After pre-training all the networks until convergence, all the networks are trained jointly with the corresponding objective functions with an early stopping strategy. We use Adam optimiser with a learning rate 3.0×10^{-6} for the pre-training, and 3.0×10^{-7} for the joint training.

4.3.6 *Adaptations for In-the-Wild Recordings*

Our framework allows finetuning the networks with 2D annotations only using the 2D reprojection loss. Such adjustment of the network weights is especially effective for in-the-wild recordings which differ from the training samples in many aspects (*e.g.* in the background, lighting conditions or camera poses). We use the estimated 2D joint keypoints from OpenPose (Cao et al., 2019, 2017; Simon et al., 2017; Wei et al., 2016) as a pseudo-ground-truth 2D annotation to train our network, see Fig. 4.8 for the results of the ablative study for visual comparisons of the results with and without finetuning.

4.3.7 *Applications*

Since our framework estimates the global translation, root orientation and joint angles, virtual characters can be directly animated using the output of our method. We can also visualise the estimated torques and ground reaction forces that explain the motion in the scene; see Fig. 4.1-(right) for an example. The purple vectors represent the estimated ground reaction forces, and the more saturated green hue on the links represents stronger torques applied on the child joints.

4.4 NETWORK DETAILS

We schematically visualise the network details in Fig. 4.4. Our implementations of $\mathcal{C}_T$ and $\mathcal{C}_P$ are based on Zou et al. (2020) and composed of 1D convolutional layers with residual blocks. We use the replication padding layer of size 1 for the embedding block and size 4 for the residual block. The kernel size of the 1D convolutional layer for the embedding and residual blocks are 3 and 5, respectively. For the 1D convolution in the residual blocks, we use the dilation of size 2. For $\mathcal{C}_T$—although it is possible to estimate $\hat{\mathbf{q}}_{rr}$ and $\mathbf{b}$ with a single neural network—we observed that estimating the global rotation, joint angles and contact labels with three different networks shows higher accuracy. Therefore, $\mathcal{C}_P$ consists of three replicated networks with the difference in the output layer, see Fig. 4.4 for the details. For GRFNet and DyNet, all the inputs

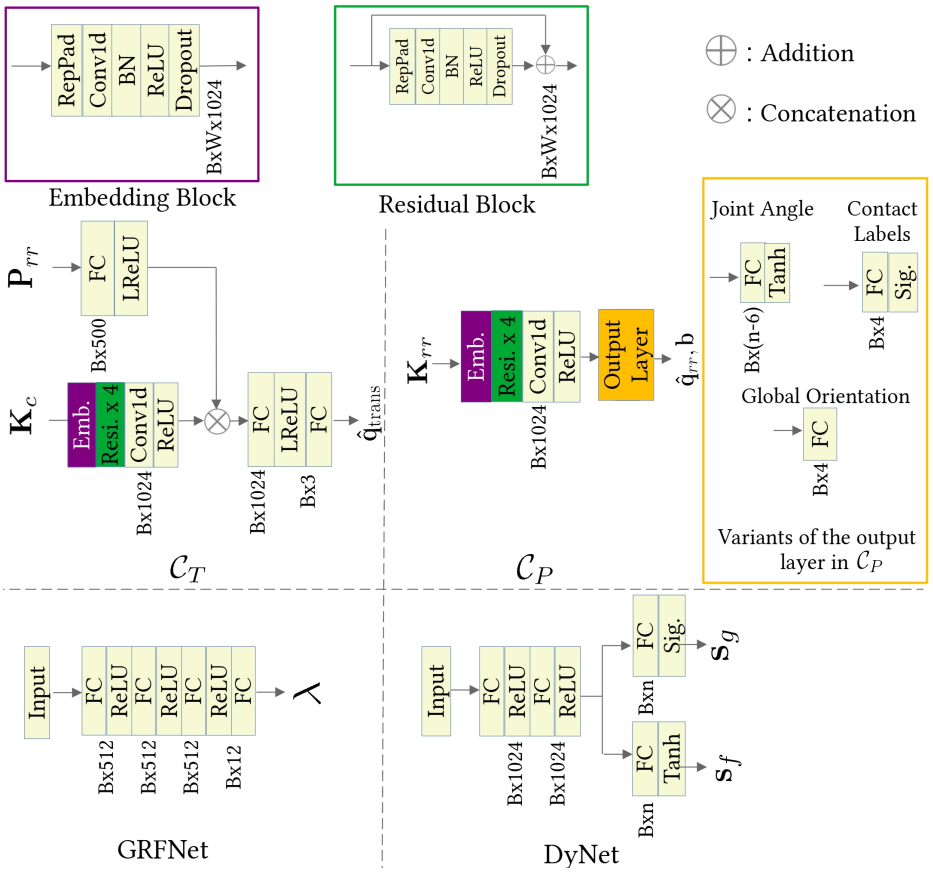


Figure 4.4: Schematic visualisations of the network details. "Emb." and "Resi." stand for the embedding block (purple box) and residual block (green box), respectively. "BN", "RepPad", "FC", "Sig." and "Conv1D" represent batch normalisation, replication padding, fully-connected layer, sigmoid function and 1D convolution, respectively. The numbers next to the layers represent the output dimensionality. "B" and "W" represent the batch size and temporal window size, respectively.

are concatenated to one vector and fed to the networks. We can estimate $\mathbf{k}_p$ and α directly by DyNet, however, similar to Chentanez et al. (2018), we obtain $\mathbf{s}_g$ and $\mathbf{s}_f$ ($0 < \mathbf{s}_g < 1$ and $-1 \leq \mathbf{s}_f \leq 1$) using sigmoid and tanh functions, and compute $\mathbf{k}_p = 2\mathbf{s}_g\mathbf{k}_p^{\text{ini}}$ and $\alpha = \gamma\mathbf{s}_f$; $\mathbf{k}_p^{\text{ini}}$ denotes the initial gain parameters which are determined following Shimada et al. (2020), and γ is the coefficient which is determined empirically. Note that we use the fixed $\mathbf{k}_p^{\text{ini}}$ and $\gamma = 10$ values through all the experiments. We observed that this formulation leads to improved stability and faster convergence of the network training than directly estimating $\mathbf{k}_p$ and α, since the network outputs are always within the normalised range.

Table 4.1: Comparisons of 3D joint position errors on DeepCap (Habermann et al., 2020), Human 3.6M (Ionescu et al., 2013) and MPI-INF-3DHP(Mehta et al., 2017a) datasets. "†" denotes physics-based algorithms, otherwise a kinematic algorithm. "∗" denotes MotioNet with causal convolutions which does not have access to the future frames, *i.e.* the similar problem set as our approach. For DeepCap dataset, the numbers on the left and right of our approach represent the 3D accuracy with and without training on DeepCap dataset, respectively.

		DeepCap			Human 3.6M			MPI-INF-3DHP		
		MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]	MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]	MPJPE↓ [*mm*]	PCK ↑ [%]	AUC ↑ [%]
Procrustes	Ours†	**52.6**/63.6	**97.3**/95.9	**67.1**/60.1	58.2	96.1	64.4	99.1	85.5	42.7
	PhysCap†	68.9	95.0	57.9	65.1	94.8	60.6	104.4	83.9	43.1
	MotioNet*	123.0	73.0	31.0	59.1	-	-	-	-	-
	VIBE	80.1	93.3	50.1	**41.5**	-	-	**63.4**	-	-
	VNect	68.4	94.9	58.3	62.7	95.7	61.9	104.5	84.1	43.2
	HMR	77.1	93.8	52.4	54.3	**96.9**	**66.6**	87.8	**87.1**	**50.9**
	HMMR	75.5	93.8	53.1	55.0	96.6	66.2	106.9	79.5	44.8
no Procrustes	Ours†	**72.7**/88.6	**92.6**/85.7	**55.3**/47.4	76.5	**89.5**	**55.0**	134.5	69.8	30.2
	PhysCap†	113.0	75.4	39.3	97.4	82.3	46.4	122.9	72.1	35.0
	MotioNet*	257.4	33.0	13.3	-	-	-	-	-	-
	VIBE	96.7	85.9	42.4	**65.9**	-	-	**97.7**	-	-
	VNect	102.4	80.2	42.4	89.6	85.1	49.0	120.2	**74.0**	**36.1**
	HMR	113.4	75.1	39.0	78.9	88.2	54.1	130.5	69.7	35.7
	HMMR	101.4	81.0	42.0	79.4	88.4	53.8	174.8	60.4	30.8

4.5 EXPERIMENTS

We evaluate our physionical approach for monocular 3D human motion capture on Human 3.6M (Ionescu et al., 2013)[2], MPI-INF-3DHP (Mehta et al., 2017a), DeepCap (Habermann et al., 2020) as well as newly recorded sequences. We first provide implementation details (Sec. 4.5.1) and then show qualitative results (Sec. 4.5.2) as well as the quantitative outcomes (Sec. 4.5.2).

4.5.1 *Implementation*

Our neural networks are implemented using PyTorch (Paszke et al., 2019) and Python 3.7. Adam optimiser was used to train them. For the computation of dynamics quantities, we use *Rigid Body Dynamics Library* (Felis, 2017). For the implementation of the differentiable optimisation

2 All experiments and training using Human 3.6M were conducted at MPII.

Table 4.2: Global 3D translation error on DeepCap dataset (Habermann et al., 2020). Note that our networks are trained on Human3.6M (Ionescu et al., 2013) and MPI-INF-3DHP (Mehta et al., 2017a), and *not* trained on DeepCap dataset (Habermann et al., 2020).

	Ours	Ours w/o C_T module	Ours w/o input cano.	PhysCap	VNect	VIBE
MPJPE [mm]↓	**62.6**	68.7	105.0	110.5	112.6	244.5

layer we use Agrawal et al. (2019a), and *Pybullet* (Coumans and Bai, 2016) for visualisation purposes. Our approach is evaluated on a workstation with 32 GB RAM, AMD EPYC 7502P 32-Core Processor and NVIDIA QUADRO RTX 8000.

4.5.2 *Quantitative Results*

In this section, we compare our method with other related kinematic-based methods, *i.e.* VNect (Mehta et al., 2017b), HMR (Kanazawa et al., 2018), HMMR (Kanazawa et al., 2019), VIBE (Kocabas et al., 2020a) and MotioNet (Shi et al., 2020), as well as the recent physics-based method PhysCap (Shimada et al., 2020) on benchmark datasets (Habermann et al., 2020; Ionescu et al., 2013; Mehta et al., 2017a).

We follow the evaluation methodology proposed in Shimada et al. (2020) which suggests comparisons of monocular 3D human motion capture using an extended set of metrics. Along with the standard root-relative 3D joint position accuracy metrics, *i.e.* mean per-joint position error (MPJPE) [mm] (the lower, the better), percentage of correct key-points (PCK) [%] and area under ROC curve (AUC) [%] (the higher, the better), we report the global 3D translation error and 2D re-projection errors by projecting the estimated 3D joints onto the input and evaluation (unseen) views. Reprojection to evaluation views reveals various effects (related to physical implausibility) which are difficult to access based only on root-relative 3D errors and reprojections to the input views. Further complementary metrics measuring the degree of plausibility of the reconstructed poses are Mean Penetration Error (MPE), Percentage of Non-Penetration (PNP) and temporal consistency error. MPE evaluates the average distance between the foot and floor when there is actually a

Table 4.3: Comparison of temporal smoothness on the DeepCap (Habermann et al., 2020) and Human 3.6M datasets (Ionescu et al., 2013).

		Ours	PhysCap	VNect	HMR	HMMR	VIBE
DeepCap	e_{smooth}	**5.8**	6.3	11.6	11.7	8.1	7.2
	σ_{smooth}	8.1	4.1	8.6	9.0	5.1	10.1
Human 3.6M	e_{smooth}	**4.5**	7.2	11.2	11.2	6.8	-
	σ_{smooth}	6.9	6.9	10.1	12.7	5.9	-

foot-floor contact in the scene (lower is better). PNP shows the ratio of no foot penetration into the floor (higher reflects a higher degree of physical plausibility).

3D JOINT POSITIONS Table 4.1 summarises the root-relative 3D joint position errors. The first and second row blocks report the calculations with and without Procrustes alignment, respectively. On Human 3.6M and MPI-INF-3DHP with Procrustes alignment, the accuracy of our method is average among the compared methods. On Human 3.6M, we obtain a slightly lower MPJPE than VNect, MotioNet and PhysCap while HMR, HMMR and VIBE achieve the lowest errors in overall. On MPI-INF-3DHP, the overall tendency is preserved, though in addition we outperform HMMR. On the DeepCap dataset, we report the results of two different variants, *i.e.* when the networks are trained on DeepCap dataset + Human3.6M + MPI-INF-3DHP (on the left) and Human3.6M + MPI-INF-3DHP without DeepCap dataset (on the right). Even without using DeepCap dataset for training, ours outperforms other tested algorithms. Compared with Human 3.6M and MPI-INF-3DHP, DeepCap dataset contains challenging motions such as dance, walking backwards, jumping and running sequences. Purely kinematic algorithms tend to fail on these challenging motions. In our case, the magnitudes of inaccuracies are regularised within a reasonable range thanks to the explicit physics model, which results in a lower MPJPE.

Most of the competing methods overfit to a single dataset and cannot generalise well to other datasets. Without Procrustes alignment, our approach outperforms all other evaluated methods on DeepCap dataset, and ranks second on Human 3.6M. We consistently outperform the most related methods on DeepCap, Human 3.6M and MPI-INF-3DHP (with Procrustes), which estimate global 3D human poses and can be directly

Table 4.4: 2D projection error of a frontal view (input) and side view (non-input) on DeepCap dataset (Habermann et al., 2020).

	Front View		Side View	
	e_{2D}^{input} [pix]	σ_{2D}^{input}	e_{2D}^{side} [pix]	σ_{2D}^{side}
Ours	**7.6**	7.5	**11.5**	**13.1**
PhysCap	21.1	6.7	35.5	16.8
VNect	14.3	2.7	37.2	18.1

used for virtual character animation. This list also includes the physics-based PhysCap, *i.e.* the most closely related method to ours. The high accuracy of purely kinematics methods (in Table 4.1, those are all methods without "†") on Human 3.6M and MPI-INF-3DHP comes at the price of frequent and sudden changes in the 3D joint positions, which result in jitters and other artefacts. Note that the obvious artefacts such as jitter and foot-floor penetration are not revealed by these conventional metrics, which suggests that considering those alone is not enough to judge the motion quality: they do not draw the complete picture, especially when having computer graphics applications in mind; hence, we report several additional metrics to provide a more comprehensive assessment of the motions.

GLOBAL TRANSLATION ERRORS We also qualitatively compare the accuracy of the global character's root position (translation) on the DeepCap dataset in Table 4.2. Note that we train our method only on Human 3.6M and MPI-INF-3DHP datasets in this experiment, which also evaluates the generalisability of the translation estimator C_T trained with the canonical 2D keypoints. We also show our ablated models 1) without the C_T module and 2) without the input canonicalisation, in the third and fourth columns, respectively. In the third case, instead of using C_T, we estimate the global translation by solving a 2D reprojection-based optimisation with gradient descent, given the estimated root-relative 3D pose and 2D joint keypoints. Without the input keypoint canonicalisation, the performance of our algorithm is significantly decreased compared to our full model. This is because the network overfits to the camera parameters which are observed in the training datasets without the input canonicalisation. For VIBE—since it does not return a global 3D

Table 4.5: Comparison of Mean Penetration Error (MPE) and Percentage of Non-Penetration (PNP) on DeepCap dataset (Habermann et al., 2020).

	MPE [mm]↓	PNP [%]↑
Ours	28.9	92.3
Ours w/o HC	29.7	89.6
PhysCap	**28.0**	**92.9**
VNect	39.3	45.6

translation—we apply re-scaling of bone lengths to match the ground-truth bone lengths and likewise solve a reprojection-based optimisation to estimate the global translation which we report in the seventh column. We see that even without C_T module activated, our method outperforms PhysCap, VNect and VIBE by 75% (for PhysCap) and more (VNect and VIBE).

PHYSICAL PLAUSIBILITY MEASUREMENT We further evaluate our approach using quantitative measures for the plausibility of the 3D motion. Table 4.3 shows the temporal smoothness error e_{smooth} which is computed as follows (Shimada et al., 2020):

$$\begin{aligned} e_{\text{smooth}} &= \frac{1}{Tk}\sum_{t=1}^{T}\sum_{s=1}^{k} \left\| \text{Jit}_{\text{GT}} - \text{Jit}_X \right| , \\ \text{with } \text{Jit}_X &= \left\| \mathbf{p}_X^{s,t} - \mathbf{p}_X^{s,t-1} \right\| \text{ and } \text{Jit}_{\text{GT}} = \left\| \mathbf{p}_{\text{GT}}^{s,t} - \mathbf{p}_{\text{GT}}^{s,t-1} \right\| , \end{aligned} \tag{4.26}$$

where $\mathbf{p}^{s,t}$ represents the 3D position of joint s in the frame t; T and k denote the total numbers of frames in the input sequence and target 3D joints, respectively. Smaller e_{smooth} means less jitter in the reconstructed 3D motions. Our approach shows the lowest e_{smooth} among all tested methods, followed by the physics-based method PhysCap and VIBE and HMMR with temporal constraints (*i.e.* these methods take several frames as inputs). This confirms the significance of our explicit physics model for more physically plausible results.

We also report in Table 4.4 the 2D reprojection error onto the input views ($e_{\text{2D}}^{\text{input}}$) and side views ($e_{\text{2D}}^{\text{side}}$) that are not used as inputs to the algorithms: $\sigma_{\text{2D}}^{\text{input}}$ and $\sigma_{\text{2D}}^{\text{side}}$ represent the standard deviation of $e_{\text{2D}}^{\text{input}}$

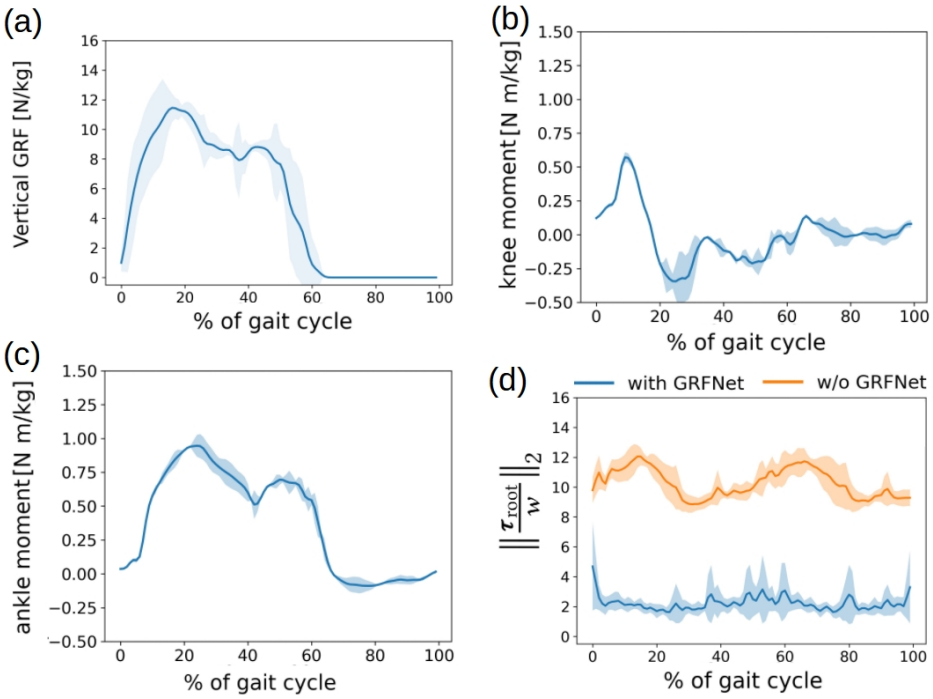


Figure 4.5: Estimated forces of the walking sequences from the DeepCap dataset. The thick line and coloured area represent the means and standard deviations, respectively. The force graph lies in the reasonable range for walking motion (*cf.* Shahabpoor and Pavic (2017) and Zell et al. (2020)), and mostly shows a smooth curve.

and e_{2D}^{side}, respectively. Reprojection onto non-input-view is an expressive operation since it reveals the artefacts which are not observable from the input view (*e.g.* body leaning and wrong translation estimation along the depth direction). Again, our results lead to the lowest metric among all methods which suggests that our global 3D motion capture is more physically plausible compared to other methods.

Finally, Table 4.5 reports the physical plausibility measurement for foot-floor penetration. Our result is on par with PhysCap which introduces a hard constraint to prevent foot-floor penetration, followed by the purely kinematic method VNect. We also show our ablated model without the hard constraint layer (Sec. 4.3.4.4). Compared to it, our full architecture shows better performance in terms of the foot-floor penetration metric.

GRF FUNCTION In Fig. 4.5, we plot the forces estimated by our physionical algorithm for the walking motion from the DeepCap dataset.

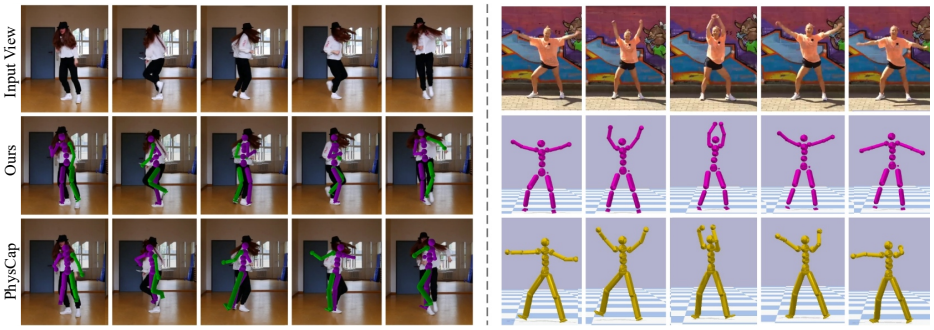


Figure 4.6: Qualitative comparisons of methods with physics-based constraints on videos with fast motions. While having a consistently improved accuracy on general motions compared to PhysCap, our approach can capture significantly faster motions as it learns motion priors and the associated gains of the neural PD controller from data.

The thick lines and coloured areas represent the mean values and standard deviations, respectively. In Figs. 4.5-(a), (b) and (c), we show the estimated GRF along the vertical direction and joint torques of knee and ankle, respectively. The curve is smooth and is in a reasonable range for walking motions. Interested readers are referred to Shahabpoor and Pavic (2017) and Zell et al. (2020) for visual comparisons with ground-truth GRF curves for an exemplary walking sequence obtained with force plates. Note that our approach accepts only a single 2D image sequence as input and does not require any ground-truth forces for its training unlike Zell et al. (2020). In Fig. 4.5-(d), we show an ablative study of GRFNet. As elaborated in Sec. 4.3.4.2, GRFNet minimises the presence of unnatural virtual forces directly applied on the character's root joint τ_{root} and tries to explain the root motion by the GRF only, as much as possible. We report $\left\| \frac{\tau_{\text{root}}}{w} \right\|_2$ for walking cycles, where w is the character's weight. Without GRFNet, the magnitude of the virtual force acting directly on the root is $\sim$5 times higher compared to the case with the former. This suggests that GRFNet helps to estimate more physically plausible forces in the proposed framework.

4.5.3 *Qualitative Results*

We further show results on multiple in-the-wild sequences. All in all, we observe that our physionical method outputs temporally consistent global 3D human poses that not only accurately project to the input views

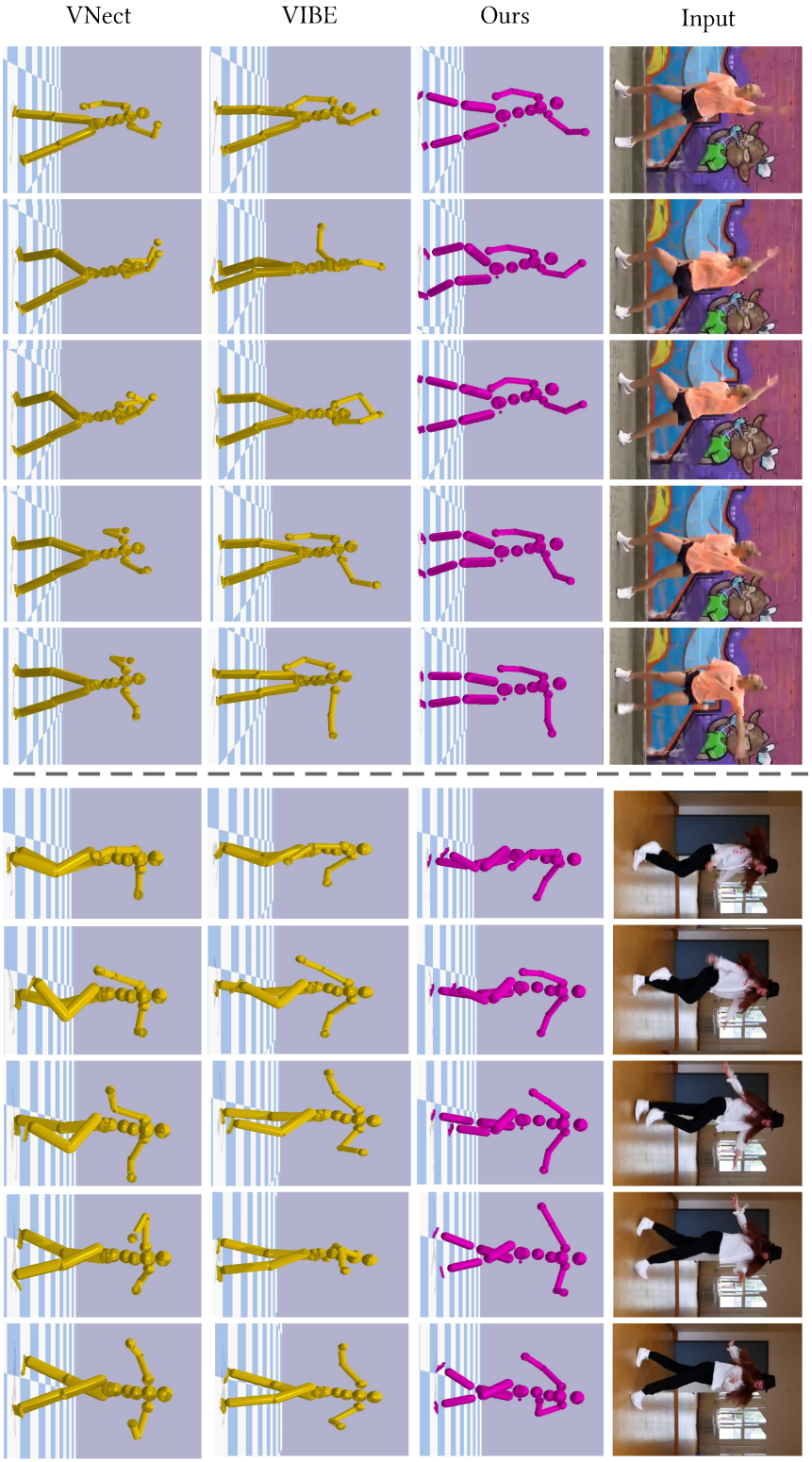


Figure 4.7: Results of our method compared to purely-kinematic methods VIBE (3D human pose and shape estimation) (Kocabas et al., 2020b) and VNect (3D human motion capture) (Mehta et al., 2017b). Our reconstructions are more temporally smooth, whereas the competing methods show frame-to-frame jitter along all axes.

but also look physically plausible when observed from arbitrary views in the 3D space. Our reconstructed 3D motions show significantly mitigated artefacts such as spurious global translational variations along the depth dimension, foot-floor penetration and jitters.

We qualitatively compare our method with the most related work PhysCap (Shimada et al., 2020) in Fig. 4.6. It is noticeable that our method catches up with fast motions with significantly mitigated motion delay thanks to the learned PD controller gain values for different motion types (see Fig. 4.6-(left)). PhysCap struggles to reconstruct correct 3D motions when fast motion appears due to its fixed gain parameters of the PD controller. Also note that our framework shows more accurate articulations on the in the-wild-sequence (see Fig. 4.6-(right)). In Fig. 4.7, we compare our method with the state-of-the-art kinematic-based methods VNect (Mehta et al., 2017b) and VIBE (Kocabas et al., 2020b) on in-the-wild sequences. Only our method reconstructs smooth sequential 3D motions. The 3D motions by VNect and VIBE show sudden changes in joint positions which are observed as jitters in the sequential visualisation.

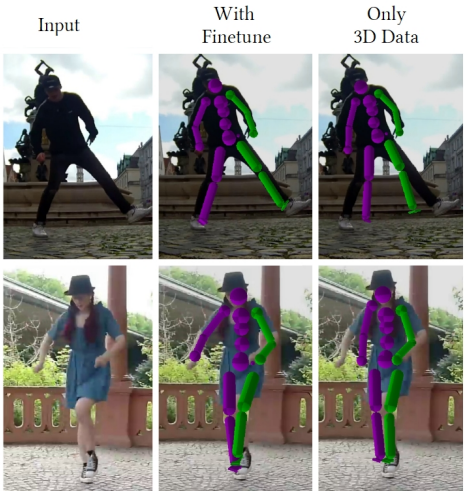


Figure 4.8: The accuracy of our method with finetuning using additional 2D annotations improves for in-the-wild sequences, compared to training using 3D data only.

We next show the results of our approach with and without finetuning our network with 2D keypoints obtained on the sequences in the wild, see Fig. 4.8 for the qualitative comparison. We use OpenPose (Cao et al., 2019) to obtain 2D keypoints, and the networks are finetuned with the 2D reprojection loss. After the finetuning, our framework shows better overlay and visually more accurate 3D motions compared to the networks trained with the 3D benchmark datasets only (Human 3.6M, MPI-INF-3DHP and DeepCap).

4.6 CONCLUSIONS

We introduced a new fully-neural approach for 3D human motion capture from monocular RGB videos with hard physics-based constraints which runs at interactive framerates and achieves state-of-the-art results on multiple metrics. Our neural physical model allows learning motion priors and the associated physical properties, as well as gain values of the neural PD controller from data. Thanks to the custom neural layer, which expresses hard physics-based constraints, our architecture is fully differentiable. In addition, it can be trained jointly on several datasets thanks to the new form of input canonicalisation. Our experiments demonstrate that compared to PhysCap—a method with physics-based boundary conditions—our physionical approach captures significantly faster motions, while being more accurate in terms of various 3D reconstruction metrics. Thanks to the full differentiability, the proposed method can be finetuned on datasets with 2D annotations only, which improves the reconstruction fidelity on in-the-wild footage. These properties make it well-suitable for direct virtual character animation from monocular videos, without requiring any further post-processing of the estimated global 3D poses.

We believe that the proposed method opens up multiple directions for future research. Our architecture can be classified as a 2D keypoint lifting approach, which has both advantages (*e.g.* the possibility of 2D keypoint normalisation, on the one hand) and downsides (*e.g.* reliance on the accuracy of 2D keypoint detectors, on the other). Next, our results naturally lead to the question of what is the most effective way to integrate physics-based boundary conditions in neural architectures, and how the proposed ideas can be applied to many related problem settings.

5

HULC: 3D HUMAN MOTION CAPTURE WITH POSE MANIFOLD SAMPLING AND DENSE CONTACT GUIDANCE

The previous chapter introduced a new fully learning-based approach for monocular RGB 3D human motion capture with explicit rigid body dynamics modelling as an intricate part of the proposed neural network architectures. While the captured motion appeared plausible and natural, interactions were restricted solely to those between the feet and the floor. However, our everyday activities involve more intricate interactions with complex environments, such as sitting on a couch or touching a table. This chapter (published as Shimada et al., 2022) presents a new formulation of RGB-based MoCap with complex scene interactions. The method consists of two key components: (1) contact-guided 3D localisation and (2) collision handling in a learned pose manifold formulated as a hard constraint. Thanks to these novel components, the reconstructed motions show much more plausible interactions with the static scene qualitatively and quantitatively in comparison to other scene-aware MoCap methods.

5.1 INTRODUCTION

3D human motion capture (MoCap) from a single colour camera received a lot of attention over the past years (Bogo et al., 2016; Chen and Ramanan, 2017; Choi et al., 2021; Habibie et al., 2019; Kanazawa et al., 2018, 2019; Kocabas et al., 2020a, 2021a,b; Kolotouros et al., 2019, 2021; Martinez et al., 2017; Mehta et al., 2017a, 2020, 2017b; Moreno-Noguer, 2017; Newell et al., 2016; Pavlakos et al., 2017, 2018b; Pavllo et al., 2019; Rhodin et al., 2018; Shi et al., 2020; Sun et al., 2019; Tekin et al., 2016; Tomè et al., 2017; Wei and Chai, 2010; Yang et al., 2018; Zhou et al., 2017). Its applications range from mixed and augmented reality, to movie production and game development, to immersive virtual communication and telepresence. MoCap techniques that not only focus on humans *in a vacuum* but also account for the scene environment—this encompasses awareness of the

Table 5.1: Overview of inputs and outputs of different methods. "τ" and "env. contacts" denote global translation and environment contacts, respectively. "$*$" stands for sparse marker contact labels.

Approach	Inputs	Outputs				
		body pose	τ	absolute scale	body contacts	env. contacts
PROX (Hassan et al., 2019)	RGB + scene mesh	✓	✓	✗	✗	✗
PROX-D (Hassan et al., 2019)	RGBD + scene mesh	✓	✓	✗	✗	✗
LEMO (Zhang et al., 2021d)	RGB(D) + scene mesh	✓	✓	✗	✓*	✗
HULC (ours)	RGB + scene point cloud	✓	✓	✓	✓	✓

physics or constraints due to the underlying scene geometry—are coming increasingly into focus (Hassan et al., 2019; Rempe et al., 2021, 2020; Shimada et al., 2021, 2020; Yi et al., 2022; Zanfir et al., 2018; Zhang et al., 2021d).

Taking into account interactions between the human and the environment in MoCap poses many challenges, as not only articulations and global translation of the subject must be accurate, but also contacts between the human and the scene need to be plausible. A misestimation of only a few parameters, such as a 3D translation, can lead to reconstruction artefacts that contradict physical reality (*e.g.* body-environment penetrations or body floating). On the other hand, known human-scene contacts can serve as reliable boundary conditions for improved 3D pose estimation and localisation. While several algorithms merely consider human interactions with a ground plane (Rempe et al., 2021, 2020; Shimada et al., 2021, 2020; Zanfir et al., 2018), a few other methods also account for the contacts and interactions with the more general 3D environment (Hassan et al., 2019; Zhang et al., 2021d). However, due to the depth ambiguity of the monocular setting, their estimated subject's root translations can be inaccurate, which can create implausible body-environment collisions. Next, they employ a body-environment collision penalty as a soft constraint. Therefore, the convergence of the optimisation to a bad local minima can also cause unnatural body-environment collisions. This chapter addresses the limitations of the current works and proposes a new 3D **HU**man MoCap framework with pose manifold samp**L**ing and guidance by body-scene **C**ontacts, abbreviated as HULC. It improves over other monocular 3D human MoCap methods that consider constraints from 3D scene priors (Hassan et al., 2019; Zhang et al., 2021d). Unlike existing

works, HULC estimates contacts not only on the human body surface but also on the environment surface for improved global 3D translation estimations. Next, HULC introduces a pose manifold sampling-based optimisation to obtain plausible 3D poses while handling the severe body-environment collisions in a *hard manner*. Our approach regresses more accurate 3D motions respecting scene constraints while requiring less-structured inputs (*i.e.* an RGB image sequence and a point cloud of the static background scene) compared to the related monocular scene-aware methods (Hassan et al., 2019; Zhang et al., 2021d) that require a complete mesh and images. HULC returns physically plausible motions, an absolute scale of the subject and dense contact labels both on a human template surface model and the environment.

HULC features several innovations which in interplay enable its functionality, *i.e.* 1) a new learned implicit function-based dense contact label estimator for humans and the general 3D scene environment, and 2) a new pose optimiser for scene-aware pose estimation based on a pose manifold sampling policy. The first component allows us to jointly estimate the subject's absolute scale and its highly accurate root 3D translations. The second component prevents severe body-scene collisions and acts as a hard constraint, in contrast to widely-used soft collision losses (Hassan et al., 2019; Mahmood et al., 2019). To train the dense contact estimation networks, we also annotate contact labels on a large-scale synthetic daily motion dataset: GTA-IM (Cao et al., 2020). To summarise, our primary technical contributions are as follows:

- A new 3D MoCap framework with simultaneous 3D human pose localisation and body scale estimation guided by estimated contacts. It is the first method that regresses the dense body and environment contact labels from an RGB sequence and a point cloud of the scene using an implicit function (Sec. 5.3.3).
- A new pose optimisation approach with a novel pose manifold sampling yielding better results by imposing hard constraints on incorrect body-environment interactions (Sec. 5.3.4).
- Large-scale body contact annotations on the GTA-IM dataset (Cao et al., 2020) that provides synthetic 3D human motions in a variety of scenes (Fig. 5.1 and Sec. 5.4).

We report quantitative results, including an ablative study, which show that HULC outperforms existing methods in 3D accuracy and on physical plausibility metrics (Sec. 5.5).

5.2 RELATED WORK

5.2.1 *Classic MoCap approaches*

Most MoCap methods estimate 3D poses alone or along with the body shape from an input image or video (Bogo et al., 2016; Chen and Ramanan, 2017; Choi et al., 2021; Habibie et al., 2019; Jiang et al., 2020; Kanazawa et al., 2018, 2019; Kocabas et al., 2020a, 2021a; Kolotouros et al., 2019, 2021; Martinez et al., 2017; Mehta et al., 2017a; Moreno-Noguer, 2017; Newell et al., 2016; Pavlakos et al., 2017, 2018b; Rhodin et al., 2018; Shi et al., 2020; Sun et al., 2019; Tekin et al., 2016; Tomè et al., 2017; Wei and Chai, 2010; Yang et al., 2018; Zhang et al., 2020b; Zhou et al., 2017). Some methods also estimate 3D translation of the subject in addition to the 3D poses (Kocabas et al., 2021b; Mehta et al., 2020, 2017b; Pavllo et al., 2019). Unlike our HULC, these methods do not consider the interactions with arbitrary scene geometries.

5.2.2 *Awareness of human-scene contacts*

Knowing contacts is helpful for the estimation and synthesis (Hassan et al., 2021a; Wang et al., 2021) of plausible 3D human motions. Some existing works regress sparse joint contacts on a kinematic skeleton (Li et al., 2019; Rempe et al., 2021, 2020; Shimada et al., 2021, 2020; Zou et al., 2020) or sparse markers (Zhang et al., 2021d). A few approaches forecast contacts on a dense human mesh surface (Hassan et al., 2021b; Müller et al., 2021). Hassan et al. (2021b) place a human in a 3D scene considering the semantic information and dense human body contact labels. Müller et al. (2021) propose a dataset with discrete annotations for self-contacts on the human body. Consequently, they apply a self-contact loss for more plausible final 3D poses. Unlike the existing works, our algorithm estimates vertex-wise dense contact labels on the human body surface from an RGB input only. Along with that, it also regresses dense

contact labels on the environment given the scene point cloud along with the RGB sequence. The simultaneous estimation of the body and scene contacts allows HULC to disambiguate the depth and scale of the subject, although only a single camera view and a single scene point cloud are used as inputs.

5.2.3 *Monocular MoCap with scene interactions*

Among the scene-aware MoCap approaches (Hassan et al., 2019; Rempe et al., 2021, 2020; Shimada et al., 2021, 2020; Zanfir et al., 2018; Zhang et al., 2021d), there are a few ones that consider human-environment interactions given a highly detailed scene geometry (Hassan et al., 2019; Li et al., 2022; Zhang et al., 2021d). PROX (PROX-D) (Hassan et al., 2019) estimates 3D motions given an RGB (RGB-D) image, along with an input geometry provided as a signed distance field (SDF). Given an RGB(D) measurement and a mesh of the environment, LEMO (Zhang et al., 2021d) also produces geometry-aware global 3D human motions with an improved motion quality characterised by smoother transitions and robustness to occlusions thanks to the learned motion priors. These two algorithms require an RGB or RGB-D sequence with SDF (a 3D scan of the scene) or occlusion masks. In contrast, our HULC requires only an RGB image sequence and a point cloud of the scene; it returns dense contact labels on 1) the human body and 2) the environment, 3) global 3D human motion with translations and 4) the absolute scale of the human body. See Table 5.1 for an overview of the characteristics. Compared to PROX and LEMO, our HULC shows significantly mitigated body-environment collisions.

5.2.4 *Sampling-based human pose tracking*

Several sampling-based human pose tracking algorithms have been proposed. Some of them utilise particle-swarm optimisation (John et al., 2010; Saini et al., 2013, 2012). Charles et al. (2013) employ Parzen windows for 2D joint tracking. Similar to our HULC, Sharma et al. (2019) generate 3D pose samples by a conditional variational autoencoder (VAE) (Sohn et al., 2015) conditioned on 2D poses. In contrast, we utilise the learned

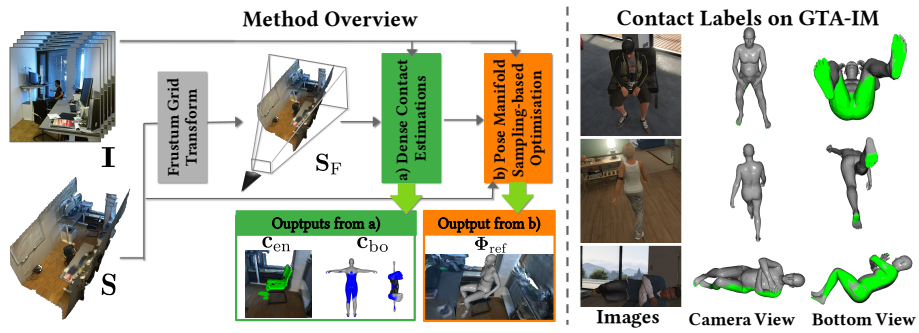


Figure 5.1: (*Left*) Given image sequence **I**, scene point cloud **S** and its associated frustum voxel grid $\mathbf{S}_F$, HULC first predicts for each frame dense contact labels on the body $\mathbf{c}_{bo}$, and on the environment $\mathbf{c}_{en}$. It then refines initial, physically-inaccurate and scale-ambiguous global 3D poses $\mathbf{\Phi}_0$ into the final ones $\mathbf{\Phi}_{ref}$ in (b). Also see Fig. 5.2 for the details of stages (a) and (b). (*Right*) Example visualisations of our contact annotations (shown in green) on GTA-IM dataset (Cao et al., 2020).

pose manifold of VAE for sampling, which helps to avoid local minima and prevent body-scene collisions. Also, unlike Sharma et al. (2019), we sample around a latent vector obtained from the VAE's encoder to obtain poses that are plausible and similar to the input 3D pose.

5.3 METHOD

Given monocular video frames and a point cloud of the scene registered to the coordinate frame of the camera, our goal is to infer physically plausible global 3D human poses along with dense contact labels on both body and environment surfaces. Our approach consists of two stages (Fig. 5.1):

- **Dense body-environment contacts estimation**: Dense contact labels are predicted on body and scene surfaces using a learning-based approach with a pixel-aligned implicit representation inspired by Saito et al. (2019) (Sec. 5.3.3);
- **Sampling-based optimisation on the pose manifold**: We combine sampling in a learned latent pose space with gradient descent to obtain the absolute scale of the subject and its global 3D pose, under guidance by predicted contacts. This approach significantly improves the accuracy of the estimated root translation and articulations, and mitigates incorrect environment penetrations (Sec. 5.3.4).

5.3.1 *Modelling and Notations*

Our method takes as input a sequence $\mathbf{I} = \{\mathbf{I}_1, ..., \mathbf{I}_T\}$ of T successive video frames from a static camera with known intrinsics ($T = 5$ in our experiments). We detect a squared bounding box around the subject and resize the cropped image region to 225×225 pixels. The background scene's geometry that corresponds to the detected bounding box is represented by a single static point cloud $\mathbf{S} \in \mathbb{R}^{M\times 3}$ composed of M points aligned in the camera reference frame in an absolute scale. To model the 3D pose and human body surface, we employ the parametric model SMPL-X (Pavlakos et al., 2019) (its gender-neutral version). This model defines the 3D body mesh as a differentiable function $\mathcal{M}(\boldsymbol{\tau}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\beta})$ of global root translation $\boldsymbol{\tau} \in \mathbb{R}^3$, global root orientation $\boldsymbol{\phi} \in \mathbb{R}^3$, pose $\boldsymbol{\theta} \in \mathbb{R}^{3K}$ of K joints and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ capturing the body's identity. For efficiency, we downsample the original SMPL-X body mesh with over 10k vertices to $\mathbf{V} \in \mathbb{R}^{N\times 3}$, where $N = 655$. In the following, we denote $\mathbf{V} = \mathcal{M}(\boldsymbol{\Phi}, \boldsymbol{\beta})$, where $\boldsymbol{\Phi} = (\boldsymbol{\tau}, \boldsymbol{\phi}, \boldsymbol{\theta})$ denotes the kinematic state of the human skeleton, from which the global positions $\mathbf{X} \in \mathbb{R}^{K\times 3}$ of the $K = 21$ joints can be derived.

5.3.2 *Frustum Grid Transform*

We conduct the transformation from the scene point cloud $\mathbf{S} \in \mathbb{R}^{M\times 3}$, defined in the camera frame, into the frustum voxel grid $\mathbf{S}_\mathrm{F} \in \mathbb{R}^{32\times 32\times 256}$ whose third dimension corresponds to the discretised depth of the 3D space. Given a vertex position $p = (x, y, z)$, *i.e.* a row of $\mathbf{S}$, in a perspective frustum space, its normalised vertex $\hat{p}$ into the cuboid space reads:

$$\hat{p} = \left(f_x \frac{x}{z}, f_y \frac{y}{z}, z\right), \tag{5.1}$$

where $f = (f_x, f_y)$ is the camera's focal length. The components of all points $\hat{p}$ are then suitably normalised and binned so as to build the binary occupancy grid $\mathbf{S}_\mathrm{F}$.

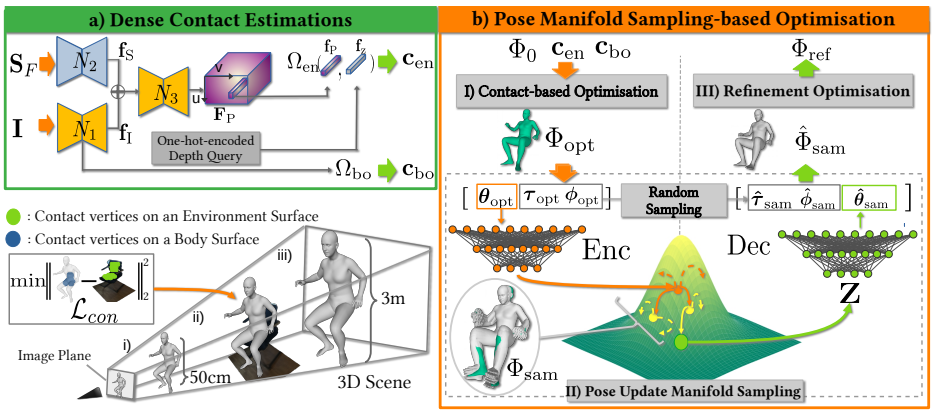


Figure 5.2: **Overview of a) dense contact estimation and b) pose manifold sampling-based optimisation.** In b-II), we first generate samples around the mapping from $\boldsymbol{\theta}_{\text{opt}}$ (orange arrows), and elite samples are then selected among them (yellow points). After resampling around the elite samples (yellow arrows), the best sample is selected (green point). The generated sample poses $\boldsymbol{\Phi}_{\text{sam}}$ (in grey colour at the bottom left in b-II)) from the sampled latent vectors are plausible and similar to $\boldsymbol{\Phi}_{\text{opt}}$. ***(bottom left of the Figure)*** Different body scale and depth combinations can be re-projected to the same image coordinates (i, ii and iii), *i.e.* **scale-depth ambiguity**. To simultaneously estimate the accurate body scale and depth of the subject (ii), we combine the body-environment contact surface distance loss $\mathcal{L}_{\text{con}}$ with the 2D reprojection loss.

5.3.3 *Contact Estimation in the Scene*

We now describe our learning-based approach for contact labels estimation on the human body and environment surfaces; see Fig. 5.2-a) for an overview of this stage. The approach takes $\mathbf{I}$ and $\mathbf{S}$ as inputs. It comprises three fully-convolutional feature extractors, N_1, N_2 and N_3, and two fully-connected layer-based contact prediction networks, Ω_{bo} and Ω_{en}, for body and environment, respectively. Network N_1 extracts from $\mathbf{I}$ a stack of visual features $\mathbf{f}_{\mathbf{I}} \in \mathbb{R}^{32\times32\times256}$. The latent space features of N_1 are also fed to Ω_{bo} to predict the vector $\mathbf{c}_{\text{bo}} \in [0,1]^N$ of per-vertex contact probabilities on the *body* surface.

We also aim at estimating the corresponding contacts on the *environment* surface using an implicit function. To train a model that generalises well, we need to address two challenges: (i) No correspondence information between the scene points and the image pixels is given; (ii) Each scene contains a variable number of points. Accordingly, we convert the scene point cloud $\mathbf{S}$ into a frustum voxel grid $\mathbf{S}_{\text{F}} \in \mathbb{R}^{32\times32\times256}$ (the third dimension corresponds to the discretised depth of the 3D space over 256 bins),

see 5.3.2 for its details. This new representation is independent of the original point-cloud size and is aligned with the camera's view direction. The latter will allow us to leverage a pixel-aligned implicit function inspired by PIFu (Saito et al., 2019), which helps the networks figure out the correspondences between pixel and geometry information. More specifically, $\mathbf{S}_F$ is fed into N_2, which returns scene features $\mathbf{f}_S \in \mathbb{R}^{32\times32\times256}$. The third encoder, N_3, ingests $\mathbf{f}_I$ and $\mathbf{f}_S$ concatenated along their third dimension and returns pixel-aligned features $\mathbf{F}_P \in \mathbb{R}^{32\times32\times64}$. Based on $\mathbf{F}_P$, Ω_{en} predicts the contact labels on the environment surface as follows. Given a 3D position in the scene, we extract the corresponding visual feature $\mathbf{f}_P \in \mathbb{R}^{64}$ at the (u, v)-position in the image space from $\mathbf{F}_P$ (via spacial bilinear interpolation), and query arbitrary depth with a one-hot vector $\mathbf{f}_z \in \mathbb{R}^{256}$. We next estimate the contact labels c_{en} as follows:

$$c_{en} = \Omega_{en}(\mathbf{f}_P, \mathbf{f}_z). \tag{5.2}$$

Given contact ground truths $\hat{\mathbf{c}}_{bo} \in \{0,1\}^N$ and $\hat{\mathbf{c}}_{en} \in \{0,1\}^M$ on the body and the environment, the five networks are trained with the following loss:

$$\mathcal{L}_{labels} = \|\mathbf{c}_{en} - \hat{\mathbf{c}}_{en}\|_2^2 + \lambda \ \mathrm{BCE}(\mathbf{c}_{bo}, \hat{\mathbf{c}}_{bo}), \tag{5.3}$$

where BCE denotes the binary cross-entropy and $\lambda = 0.3$. We use BCE for the body because the ground-truth contacts on its surface are binary; the ℓ_2 loss is used for the environment, as sparse ground-truth contact labels are smoothed with a Gaussian kernel to obtain continuous signals. For further discussions of (5.3), please see Sec. 5.3.5. At test time, we only provide the 3D vertex positions of the environment to $\Omega_{en}(\cdot)$—to find the contact area on the scene point cloud—rather than all possible 3D sampling points as queries. This significantly accelerates the search of environmental contact labels while reducing the number of false-positive contact classifications.

5.3.4 *Pose Manifold Sampling-based Optimisation*

In the second stage of the approach, we aim at recovering an accurate global 3D trajectory of the subject as observed in the video sequence; see Fig. 5.2-(b) for the overview. An initial estimate $\mathbf{\Phi}_0$ is extracted for each input image using SMPLify-X (Pavlakos et al., 2019). Its root translation

$\boldsymbol{\tau}$ being subject to scale ambiguity, we propose to estimate it more accurately, along with the actual scale h of the person with respect to the original body model's height, under the guidance of the predicted body-environment contacts (**Contact-based Optimisation**). We then update the body trajectory and articulations in the scene while mitigating the body-environment collisions with a new sampling-based optimisation on the pose manifold (**Sampling-based Trajectory Optimisation**). A subsequent refinement step yields the final global physically plausible 3D motions.

I) Contact-based Optimisation Scale ambiguity is inherent to a monocular MoCap setting: Human bodies with different scale and depth combinations in 3D can be reprojected on the same positions in the image frame; see Fig. 5.2 for the schematic visualisation. Most existing algorithms that estimate global 3D translations of a subject either assume its known body scale (Dabral et al., 2021; Shimada et al., 2021, 2020) or use a statistical average body scale (Mehta et al., 2017b). In the latter case, the estimated $\boldsymbol{\tau}$ is often inaccurate and causes physically implausible body-environment penetrations. In contrast to the prior art, we simultaneously estimate $\boldsymbol{\tau}$ and h by making use of the body-environment dense contact labels from the previous stage (Sec. 5.3.3).

For the given frame at time $t \in [\![1, T]\!]$, we select the surface regions with $\mathbf{c}_{\text{en}} > 0.5$ and $\mathbf{c}_{\text{bo}} > 0.5$ as effective contacts and leverage them in our optimisation. Let us denote the corresponding index subsets of body vertices and scene points by $\mathcal{C}_{\text{bo}} \subset [\![1, N]\!]$ and $\mathcal{C}_{\text{en}} \subset [\![1, M]\!]$. The objective function for contact-based optimisation is defined as:

$$\mathcal{L}_{\text{opt}}(\boldsymbol{\tau}, h) = \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} + \lambda_{\text{con}}\mathcal{L}_{\text{con}}, \tag{5.4}$$

where the reprojection $\mathcal{L}_{\text{2D}}$, the temporal smoothness $\mathcal{L}_{\text{smooth}}$ and the contact $\mathcal{L}_{\text{con}}$ losses weighted by empirically-set multipliers λ_{2D}, λ_{smooth} and λ_{con}, read:

$$\mathcal{L}_{\text{2D}} = \frac{1}{K}\sum_{k=1}^{K} w_k \|\Pi(\mathbf{X}_k) - \mathbf{p}_k\|_2^2, \tag{5.5}$$

$$\mathcal{L}_{\text{smooth}} = \left\|\boldsymbol{\tau} - \boldsymbol{\tau}_{\text{prev}}\right\|_2^2, \tag{5.6}$$

$$\mathcal{L}_{\text{con}} = \sum_{n \in \mathcal{C}_{\text{bo}}} \min_{m \in \mathcal{C}_{\text{en}}} \|\mathbf{V}_n - \mathbf{P}_m\|_2^2, \tag{5.7}$$

where $\mathbf{p}_k$ and w_k are the 2D detection in the image of the k-th body joint and its associated confidence, respectively, obtained by OpenPose (Cao et al., 2019); $\Pi(\cdot)$ is the perspective projection operator; $\boldsymbol{\tau}_{\text{prev}}$ is the root translation estimated in the previous frame; $\mathbf{X}_k$, $\mathbf{V}_n$ and $\mathbf{P}_m$ are, respectively, the k-th 3D joint, the n-th body vertex ($n \in \mathcal{C}_{\text{bo}}$) and the m-th scene point ($m \in \mathcal{C}_{\text{en}}$). Note that the relative rotation and pose are taken from $\boldsymbol{\Phi}_0$. The body joints and vertices are obtained from $\mathcal{M}$ using $\boldsymbol{\tau}$ and scaled with h. For $\mathcal{L}_{\text{con}}$, we use a directed Hausdorff measure (Knauer et al., 2009) as a distance between the body and environment contact surfaces. The combination of $\mathcal{L}_{\text{con}}$ and $\mathcal{L}_{\text{2D}}$ is key to disambiguate $\boldsymbol{\tau}$ and h (thus, resolving the monocular scale ambiguity). As a result of optimising (5.4) in frame t, we obtain $\boldsymbol{\Phi}^t_{\text{opt}}$, *i.e.* the global 3D human motion with absolute body scale. We solve jointly on T frames and optimise for a single h for them.

II-a) Sampling-based Trajectory Optimisation Although the poses $\boldsymbol{\Phi}^t_{\text{opt}}$, $t = 1 \cdots T$, estimated in the previous step yield much more accurate $\boldsymbol{\tau}$ and h compared to existing monocular RGB-based methods, incorrect body-environment penetrations are still observable. This is because the gradient-based optimisation often gets stuck in bad local minima. To overcome this problem, we introduce an additional sampling-based optimisation that imposes hard penetration constraints, thus significantly mitigating physically implausible collisions. The overview of this algorithm is as follows: (i) For each frame t, we first draw candidate poses around $\boldsymbol{\Phi}^t_{\text{opt}}$ with a sampling function $\mathcal{G}$; (ii) The quality of these samples is ranked by a function $\mathcal{E}$ that allows selecting the most promising ("elite") ones; samples with severe collisions are discarded; (iii) Using $\mathcal{G}$ and $\mathcal{E}$ again, we generate and select new samples around the elite ones. The details of these steps, $\mathcal{E}$ and $\mathcal{G}$, are elaborated next (dropping time index t for simplicity).

II-b) Generating Pose Samples We aim to generate N_{sam} sample states $\boldsymbol{\Phi}_{\text{sam}}$ around the previously-estimated $\boldsymbol{\Phi}_{\text{opt}} = (\boldsymbol{\tau}_{\text{opt}}, \phi_{\text{opt}}, \boldsymbol{\theta}_{\text{opt}})$. However, naïvely generating the relative pose $\boldsymbol{\theta}_{\text{sam}}$ in the same way around $\boldsymbol{\theta}_{\text{opt}}$ is highly inefficient because (i) the body pose is high-dimensional and (ii) the randomly-sampled poses are not necessarily plausible. These reasons lead to an infeasible amount of generated samples required to find a plausible collision-free pose; which is intractable on standard graphics hardware. To tackle these issues, we resort to the pose manifold learned

by VPoser (Pavlakos et al., 2019), which is a VAE (Kingma and Welling, 2014) trained on AMASS (Mahmood et al., 2019), *i.e.* a dataset with many highly accurate MoCap sequences. Sampling is conducted in this VAE's latent space rather than in the kinematics pose space. Specifically, we first map $\boldsymbol{\theta}_{\text{opt}}$ into a latent pose vector with the VAE's encoder $\text{Enc}(\cdot)$. Next, we sample latent vectors using a Gaussian distribution centred at this vector, with standard deviation $\boldsymbol{\sigma}$ (see Fig. 5.2-b). Each latent sample is then mapped through VAE's decoder $\text{Dec}(\cdot)$ into a pose that is combined with the original one on a per-joint basis. The complete sampling process reads:

$$\mathbf{Z} \sim \mathcal{N}(\,\text{Enc}(\boldsymbol{\theta}_{\text{opt}}), \boldsymbol{\sigma}), \quad \boldsymbol{\theta}_{\text{sam}} = \mathbf{w} \circ \boldsymbol{\theta}_{\text{opt}} + (1 - \mathbf{w}) \circ \text{Dec}(\mathbf{Z}), \tag{5.8}$$

where $\circ$ denotes Hadamard matrix product and $\mathbf{w} \in \mathbb{R}^{3K}$ is composed of the detection confidence values w_k, $k = 1 \cdots K$, obtained from OpenPose, each appearing three times (for each DoF of the joint). This confidence-based strategy allows weighting higher the joint angles obtained by sampling, if the image-based detections are less confident (*e.g.* under occlusions). Conversely, significant modifications are not required for the joints with high confidence values.

Since the manifold learned by VAE is smooth, the poses derived from the latent vectors sampled around $\text{Enc}(\boldsymbol{\theta}_{\text{opt}})$ should be close to $\boldsymbol{\theta}_{\text{opt}}$. Therefore, we empirically set $\boldsymbol{\sigma}$ to a small value (0.1). Compared to the naïve random sampling in the joint angle space, whose generated poses are not necessarily plausible, this pose sampling on the learned manifold significantly narrows down the solution space. Hence, a lot fewer samples are required to escape local minima. At the bottom left of Fig. 5.2-b contains examples (grey colour) of $\boldsymbol{\Phi}_{\text{sam}}$ ($N_{\text{sam}} = 10$) overlayed onto $\boldsymbol{\Phi}_{\text{opt}}$ (green).

To generate samples $(\boldsymbol{\tau}_{\text{sam}}, \phi_{\text{sam}})$ for the root translation and orientation, we generate random samples around the initial translation $\boldsymbol{\tau}_{\text{opt}}$ and ϕ_{opt} since they have only 3 DoF for each. Specifically, we generate samples by adding the randomly generated offsets $\Delta\boldsymbol{\tau} = \psi\varphi_{\boldsymbol{\tau}}$ and $\Delta\phi = \psi\varphi_{\phi}$ to $\boldsymbol{\tau}_{\text{opt}}$ and ϕ_{opt}, respectively; ψ is initialised to 1.0, and incremented by 1.0 when the solution is not found due to the hard collision constraint; $\varphi_{\boldsymbol{\tau}} \in [-0.03, 0.03]^3$ and $\varphi_{\phi} \in [-0.01, 0.01]^3$ are the values generated uniformly at random. The range of φ_{ϕ} is kept small since even a small

change of the root orientation greatly modifies the 3D joint positions. In the following, we refer to this sampling process as function $\mathcal{G}(\cdot)$.

II-c) Sample Selection The quality of the N_{sam} generated samples $\mathbf{\Phi}_{\text{sam}}$ is evaluated using the following cost function:

$$\mathcal{L}_{\text{sam}} = \mathcal{L}_{\text{opt}} + \lambda_{\text{sli}}\mathcal{L}_{\text{sli}} + \lambda_{\text{data}}\mathcal{L}_{\text{data}}, \tag{5.9}$$

$$\mathcal{L}_{\text{sli}} = \left\|\mathbf{V}_{\text{c}} - \mathbf{V}_{\text{c,pre}}\right\|_2^2, \tag{5.10}$$

$$\mathcal{L}_{\text{data}} = \left\|\mathbf{\Phi}_{\text{sam}} - \mathbf{\Phi}_{\text{opt}}\right\|_2^2, \tag{5.11}$$

where $\mathcal{L}_{\text{sli}}$ and $\mathcal{L}_{\text{data}}$ are contact sliding loss and data loss, respectively, and $\mathcal{L}_{\text{opt}}$ is the same as in (5.4) with the modification that the temporal consistency (5.6) applies to the whole $\mathbf{\Phi}_{\text{sam}}$; $\mathbf{V}_{\text{c}}$ and $\mathbf{V}_{\text{c,pre}}$ are the body contact vertices (with vertex indices in $\mathcal{C}_{\text{bo}}$) and their previous positions, respectively.

Among the N_{sam} samples ordered according to their increasing $\mathcal{L}_{\text{sam}}$ values, the selection function $\mathcal{E}_U(\cdot)$ first discards those causing stronger penetrations (in the sense that the amount of scene points inside a human body is above a threshold γ) and returns U first samples from the remaining ones. If no samples pass the collision test, we regenerate the set of N_{sam} samples. This selection mechanism introduces the collision handling in a hard manner. After applying $\mathcal{E}_U(\cdot)$, with $U<N_{\text{sam}}$, U elite samples are retained. Then, $\lfloor N_{\text{sam}}/U \rfloor$ new samples are regenerated around every elite sample using $\mathcal{G}$. Among those, the one with minimum $\mathcal{L}_{\text{sam}}$ value is retained as the final estimate. The sequence of obtained poses is temporally smoothed by Gaussian filtering to further remove jittering, which yields the global 3D motion $(\hat{\mathbf{\Phi}}^t_{\text{sam}})_{t=1}^T$ with significantly mitigated collisions.

III) Final Refinement From the previous step, we obtained the sequence $\hat{\mathbf{\Phi}}_{\text{sam}} = (\hat{\boldsymbol{\tau}}_{\text{sam}}, \hat{\phi}_{\text{sam}}, \hat{\boldsymbol{\theta}}_{\text{sam}})$ of kinematic states whose severe body-environment collisions are prevented as hard constraints. Starting from these states as initialisation, we perform a final gradient-based refinement using cost function $\mathcal{L}_{\text{sam}}$ with $\hat{\mathbf{\Phi}}_{\text{sam}}$ replacing $\mathbf{\Phi}_{\text{opt}}$. The final sequence is denoted $(\mathbf{\Phi}^t_{\text{ref}})_{t=1}^T$.

5.3.5 *Network Details*

We elaborate here on the network architectures in the dense contact estimation stage. Networks N_1 and N_3 consist of 2D-convolution-based encoder and decoder architectures. We employ Resnet-18 (He et al., 2016) for the encoder of N_1 without the last two layers, *i.e.* a fully connected layer and an average-pooling layer. We employ a U-Net (Ronneberger et al., 2015)-based architecture for N_3 with 2 sets of down-convolution and up-convolution blocks. Network Ω_{en} consists of 3 fully-connected layers with LeakyReLU (Maas et al., 2013) activation function. At the output layer, we use a sigmoid function instead of LeakyReLU. For the details of N_2, Ω_{bo} and the decoder of N_1, please see Fig. 5.3.

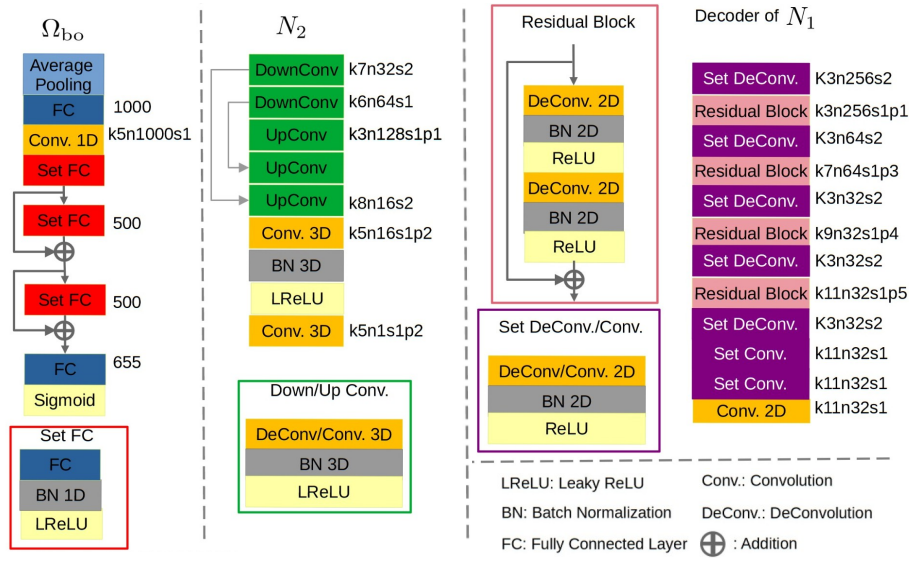


Figure 5.3: The detailed network architectures for N_2, Ω_{bo} and the decoder of N_1. The numbers next to the fully connected layers represent the output dimensionality. The numbers next to the convolution layers represent kernel size ('k'), number of kernels ('n'), size of sliding ('s') and padding size ('p'). Note that when the padding size is not shown, no padding is applied at the convolution layer.

Why this Architecture Design? Here, we discuss the architecture design choice for the environment contact estimation networks. Instead of the pixel-aligned network Ω_{en}, a 3D-convolution-based network can also be applied to obtain the voxel grid that contains per-voxel contact labels of the 3D scene. However, we observed that the 3D-convolution-based

classifier network suffers from the underfitting issue during the training due to the very small number of ground-truth positive contact labels over the total number of voxels in the grid. With the pixel-aligned implicit field, we can adjust these *unbalanced* positive and negative contact labels by manipulating the sampling points in the 3D scene which we can freely control. Also, unlike the original work (Saito et al., 2019) that provides the scalar value as a depth query, we provide a one-hot vector as a depth query to Ω_{en}: we observed that it significantly reduces the loss value during the training compared to providing the scalar depth queries.

Loss Function Design (Eq.(5.3)) Binary GT environment contacts label ('1': contact, 'o': no contact) are very sparse signals, *i.e.* only a small number of voxels (~0.01%) contain '1'. This reduces the network training stability. We observed that smoothing the environment contact labels mitigates the imbalance and enhances the training stability. Hence, with smoothing, L2-loss (not BCE) for the environment contact estimation is used. Contact labels for the body are more balanced compared to the environmental contacts. Therefore, we do not smooth them and use BCE loss.

5.4 DATASETS WITH CONTACT ANNOTATIONS

As there are no publicly available large-scale datasets with images and corresponding human-scene contact annotations, we annotate several existing datasets.

GTA-IM (Cao et al., 2020) dataset contains various daily 3D motions. First, we fit SMPL-X model onto the 3D joint trajectories in GTA-IM. For each frame, we select contact vertices on the human mesh if: i) The Euclidean distance between the human body vertices and the scene vertices is smaller than a certain threshold; ii) The velocity of the vertex is lower than a certain threshold. In total, we obtain the body surface contact annotations on $320k$ frames, see Fig. 5.1 for examples of the annotated contact labels.

PROX dataset (Hassan et al., 2019) contains scanned scene meshes, scene SDFs, RGB-D sequences, 3D human poses and shapes generated by fitting SMPL-X model onto the RGB-D sequences (considering collisions). We consider the body vertices, whose SDF values are lower than 5 cm, as

contacts. We annotate the environment contacts by finding the vertices that are the nearest to the body contacts.

GPA dataset (Wang et al., 2022b, 2020b) contains multi-view image sequences of people interacting with various rigid 3D geometries, accurately reconstructed 3D scenes and 3D human motions obtained from VICON system (*Vicon blade* n.d.) with 28 calibrated cameras. We fit SMPL-X on GPA to obtain the 3D shapes and compute the scene's SDFs to run other methods (Hassan et al., 2019, 2021b; Zhang et al., 2021d).

We extract 14 test sequences with 5 different subjects from **GPA**. We also split **PROX** (Hassan et al., 2019) into training and test sequences. The training sequences of **PROX** and **GTA-IM** (Cao et al., 2020) are used to train the contact estimation networks.

5.5 EVALUATIONS

We compare our HULC with the most related scene-aware 3D MoCap algorithms, *i.e.* PROX (Hassan et al., 2019), PROX-D (Hassan et al., 2019), POSA (Hassan et al., 2021b) and LEMO (Zhang et al., 2021d). We also test SMPLify-X (Pavlakos et al., 2019) which does not use scene constraints. The root translation of SMPLify-X is obtained from its estimated camera poses as done in Hassan et al. (2019). To run LEMO (Zhang et al., 2021d) on the RGB sequence, we use SMPLify-X (Pavlakos et al., 2019) to initialise it; we call this combination "LEMO (RGB)".

We use the selected test sequences of GPA (Wang et al., 2022b, 2020b) and PROX (Hassan et al., 2019) datasets for the quantitative and qualitative comparisons. To avoid redundancy, we downsample all the predictions to 10 fps except for the temporal consistency measurement (e_{smooth} in Table 5.4). Since the 3D poses in PROX dataset are prone to inaccuracies due to their human model fitting onto the RGB-D sequence, we use it only for reporting the body-scene penetrations (Table 5.4) and for qualitative comparisons.

5.5.1 *Implementations and Training Details*

The neural networks are implemented with PyTorch (Paszke et al., 2019) and Python 3.7. We conducted the evaluations on a computer with one

AMD EPYC 7502P 32 Core Processor and one NVIDIA QUADRO RTX 8000 graphics card. The training of the contact classification networks continued until the loss convergence using Adam optimiser (Kingma and Ba, 2015a) with a learning rate 3.0×10^{-4}. Our framework runs with 25 seconds per frame excepting the computation time of SMPLify-X (Pavlakos et al., 2019) which we use for the initial root-relative pose estimation. For the optimisation in Eq. (5.4) we use the weights $\lambda_{2D} = 1.0$, $\lambda_{smooth} = 0.01$ and $\lambda_{con} = 0.01$. For Eq. (5.9), λ_{sli} and λ_{data} are set to 0.05 and 0.1. In the final refinement optimisation step, we use $\lambda_{data} = 1.0$ while keeping the same weights for the other terms. Rather than using a Chamfer loss for $\mathcal{L}_{con}$ to minimise the body-environment contact vertex distance, we use the Hausdorff measure (Knauer et al., 2009); indeed, we observed that, with this measure, the reconstructed 3D motion is more robust to the false positive contact labels on the environment vertices. Note that the 2D keypoints are normalised by the image size. The joint angles are defined in radians.

For the evaluations, we first pre-train our networks on the whole GTA-IM dataset (Cao et al., 2020) using the image sequences and our body contact annotations. Lastly, we train our networks on PROX dataset (Hassan et al., 2019) with the environment contact labels obtained by us (see Sec. 5.4). During the training, the ground-truth scene contact vertex information is once converted into the frustum voxel grid representations as described in Sec. 5.3.2.

5.5.2 *Quantitative Results*

We report 3D joint and vertex errors (Table 5.2), global translation and body scale estimation errors (Table 5.3), body-environment penetration and smoothness errors (Table 5.4) and ablations on the sampling-based optimisation component, *i.e.* a) Manifold sampling vs. random sampling and b) Different number of sampling iterations in Fig. 5.4. "Ours (w/o S)" represents our method without the sampling optimisation component, *i.e.* only the contact-based optimisation and refinement are applied (see Fig. 5.2-(b) and Sec. 5.3.4). "Ours (w/o R)" represents our method without the final refinement. "Ours (w/o SR)" denotes ours without the sampling and refinement.

Table 5.2: Comparisons of 3D error on GPA dataset (Wang et al., 2022b, 2020b). "†" denotes that the occlusion masks for LEMO(RGB) were computed from GT 3D human mesh.

	No Procrustes			Procrustes		
	MPJPE↓ [*mm*]	PCK ↑ [%]	PVE ↓ [*mm*]	MPJPE↓ [*mm*]	PCK ↑ [%]	PVE ↓ [*mm*]
Ours	**217.9**	**35.3**	**214.7**	**81.5**	**89.3**	**72.6**
Ours (w/o S)	221.3	34.5	217.2	82.6	**89.3**	73.1
Ours (w/o R)	240.8	31.9	237.3	83.1	86.6	73.6
Ours (w/o SR)	251.1	31.5	245.2	83.9	86.6	74.1
SMPLify-X (Pavlakos et al., 2019)	550.0	10.0	549.1	84.7	85.9	74.1
PROX (Hassan et al., 2019)	549.7	10.1	548.7	84.6	86.0	73.9
POSA (Hassan et al., 2021b)	552.2	10.1	550.9	85.5	85.6	74.5
LEMO (RGB) (Zhang et al., 2021d)	570.1	8.75	570.5	83.0	86.4	73.7
LEMO (RGB) (Zhang et al., 2021d)†	570.0	8.77	570.4	83.0	86.4	73.6

3D Joint and Vertex Errors Table 5.2 compares the accuracy of 3D joint and vertex positions with and without Procrustes alignment. LEMO also requires human body occlusion masks on each frame. We compute them using the scene geometry and SMPLify-X (Pavlakos et al., 2019) results. We also show another variant "LEMO (RGB)†" whose occlusion masks are computed using the ground-truth global 3D human mesh instead of SMPLify-X. Here, we report the standard 3D metrics, *i.e.* mean per joint position error (MPJPE), percentage of correct keypoints (PCK) (@150mm) and mean per vertex error (PVE). Lower MPJPE and PVE represent more accurate 3D reconstructions, higher PCK indicates more accurate 3D joint positions.

On all these metrics, HULC outperforms other methods both with and without Procrustes. Notably, thanks to substantially more accurate global translations obtained from the contact-based optimisation (Sec. 5.3.4), HULC significantly reduces the MPJPE and PVE with a big margin, *i.e.* $\approx$60% error deduction in MPJPE and PVE w/o Procrustes compared to the second-best method. The ablative studies on Table 5.2 also indicate that both the sampling and refinement optimisations contribute to accurate 3D poses. Note that the sampling optimisation alone ("Ours (w/o R)") does not significantly reduce the error compared to "Ours (w/o SR)". This is because the sampling component prioritises the removal of envi-

Table 5.3: Ablations and comparisons for global translations and absolute body length on GPA dataset.

	global translation error [m] ↓	absolute bone length error [m]↓
Ours (+1m)	**0.242**	0.104
Ours (+3m)	0.244	**0.097**
Ours (+10m)	0.244	0.109
Baseline (+1m)	0.751	0.498
Baseline (+3m)	1.033	0.560
Baseline (+10m)	2.861	1.918
SMPLify-X (Pavlakos et al., 2019)	0.527	0.156
PROX (Hassan et al., 2019)	0.528	0.160
POSA (Hassan et al., 2021b)	0.545	0.136

ronment penetrations by introducing hard collision handling, which is the most important feature of this component. Therefore, the sampling component significantly contributes to reducing the environment collision as can be seen in Table 5.4 (discussed in the later paragraph). Applying the refinement after escaping from severe penetrations by the sampling optimisation further increases the 3D accuracy ("Ours" in Table 5.2) while significantly mitigating physically implausible body-environment penetrations (Table 5.4).

Global Translation and Body Scale Estimation Table 5.3 reports global translation and body scale estimation errors for the ablation study of the contact-based optimisation (Sec. 5.3.4). More specifically, we evaluate the output Φ_{opt} obtained from the contact-based optimisation denoted "ours". We also show the optimisation result without using the contact loss term (5.7) ("Baseline"). The numbers next to the method names represent the initialisation offset from the ground-truth 3D translation position (*e.g.* "+10m" indicates that the initial root position of the human body was placed at 10 meters away along the depth direction from the ground-truth root position when solving the optimisations).

Without the contact loss term—since global translation and body scale are jointly estimated in the optimisation—the baseline method suffers from *up-to-scale* issue (see Fig. 5.2). Hence, its results are significantly

Table 5.4: Comparisons of physical plausibility measures on GPA dataset (Wang et al., 2022b, 2020b) and PROX dataset (Hassan et al., 2019).

		GPA Dataset		PROX Dataset
		non penet.↑ [%]	e_{smooth}↓	non penet.↑ [%]
RGB	Ours	**99.4**	20.2	97.0
	Ours (w/o S)	97.6	28.1	93.8
	Ours (w/o R)	**99.4**	24.7	**97.1**
	Ours (w/o SR)	97.6	47.1	93.8
	SMPLify-X (Pavlakos et al., 2019)	97.7	43.3	88.9
	PROX (Hassan et al., 2019)	97.7	43.2	89.8
	LEMO (RGB) (Zhang et al., 2021d)	97.8	**19.9**	-
	POSA (Hassan et al., 2021b)	98.0	47.0	93.0
RGB-D	PROX-D (Hassan et al., 2019)	-	-	94.2
	LEMO (Zhang et al., 2021d)	-	-	96.4

worse due to worse initialisations. In contrast, our contact-based optimisation disambiguates the scale and depth by localising the contact positions on the environment, which confirms HULC to be highly robust to bad initialisations. Compared to the RGB-based methods PROX, POSA and SMPLify-X, our contact-based optimisation result has $\approx 40\%$ smaller error in the absolute bone length, and $\approx 57\%$ smaller error in global translation, which also contributes to the reduced body-environment collisions as demonstrated in Table 5.4 (discussed in the next paragraph).

Plausibility Measurements We also report the plausibility of the reconstructed 3D motions in Table 5.4. *Non penet.* measures the average ratio of non-penetrating body vertices into the environment over all frames. A higher value denotes fewer body-environment collisions in the sequence; e_{smooth} measures the temporal smoothness error proposed in Shimada et al. (2020). Lower e_{smooth} indicates more temporally smooth 3D motions. On both GPA and PROX datasets, our full framework mitigates the collisions thanks to the manifold sampling-based optimisations (ours vs. ours (w/o S)). It also does so when compared to other related works as well. Notably, HULC shows the least amount of collisions even compared with RGBD-based methods on the PROX dataset. Finally, the proposed method

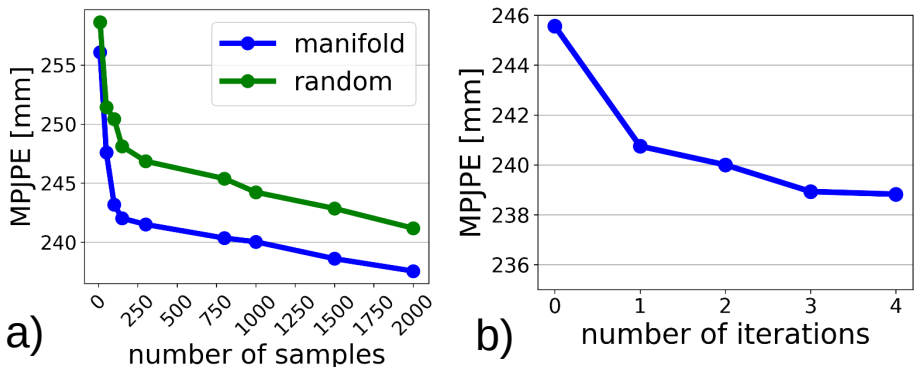


Figure 5.4: (a) MPJPE [mm] comparison with different numbers of samples for the learned manifold sampling strategy vs. the naïve random sampling in the joint angle space of the kinematic skeleton. (b) MPJPE [mm] comparison with different numbers of iterations in the sampling strategy.

also shows the significantly low e_{smooth} (on par with LEMO(RGB)) in this experiment.

More Ablations on Sampling-based Optimisation In addition to the ablation studies reported in Tables 5.2, 5.3 and 5.4, we further assess the performance of the pose update manifold sampling step (Fig. 5.2-(b)-(II)) on GPA dataset (Wang et al., 2022b, 2020b), reporting the 3D error (MPJPE [mm]) measured in world frame. Note that we report MPJPE without the final refinement step to assess the importance of the manifold sampling approach. In Fig. 5.4-(a), we show the influence of the number N_{sam} of samples on the performance of our manifold sampling strategy vs. a naïve random sampling with a uniform distribution in a kinematic skeleton frame. Specifically, for the naïve random sampling, we use the random sampling for the pose parameter $\boldsymbol{\theta}_{\text{opt}} \in \mathbb{R}^{3K}$ similar to the method explained in Sec. 5.3.4: the randomly generated offsets $\Delta\boldsymbol{\theta} = \psi\varphi_{\boldsymbol{\theta}}$ are added to $\boldsymbol{\theta}_{\text{opt}}$ to generate the pose samples; $\varphi_{\boldsymbol{\theta}} \in [-0.26, 0.26]^{3K}$ are the values that are uniformly generated at random. In Fig. 5.4-(a), since the generated samples of the learned manifold return plausible pose samples, our pose manifold sampling strategy requires significantly fewer samples compared to the random sampling ($\sim$15$\times$ more samples are required for the random sampling to reach 243 [mm] error in MPJPE). This result strongly supports the importance of the learned manifold sampling. No more than 2000 samples can be generated due to the hardware memory capacity. In Fig. 5.4-(b), we report the influence of

Table 5.5: Ablation study for the sliding loss term $\mathcal{L}_{\text{sli}}$.

	MPJPE [mm] ↓	sliding error [mm] ↓
Ours	**217.9**	**16.0**
Ours (w/o $\mathcal{L}_{\text{sli}}$)	220.2	18.5

the number of generation-selection steps using functions $\mathcal{G}$ and $\mathcal{E}_U$ (with $U = 3$) introduced in Sec. 5.3.4, with $N_{\text{sam}} = 1000$ samples. No iteration stands for choosing the best sample from the first generated batch (hence no resampling), while one iteration is the variant described in Sec. 5.3.4. This first iteration sharply reduces the MPJPE, while the benefit of the additional iterations is less pronounced. Based on these observations, we use only one re-sampling iteration with 1000 samples in the previous experiments. Finally, we ablate the confidence value-based pose merging in Eq. (5.8), setting N_{sam}=1000 and the number of iterations to 0. The measured MPJPE for with and without this confidence merging are 245.5 and 249.1, respectively.

More Ablations on $\mathcal{L}_{\text{sli}}$ For the completeness, we report the ablative study for the sliding loss term $\mathcal{L}_{\text{sli}}$ (Eq.(5.9)) used in our optimisations. In Table 5.5, we report MPJPE and sliding error $e_{\text{sli.}}$ measured in a world frame for our full framework ("Ours") and our framework w/o the sliding loss term ("Ours (w/o $\mathcal{L}_{\text{sli}}$)"). The sliding error $e_{\text{sli.}}$ is measured by computing the average of the drift of the contact vertex on the human surface, based on the assumption that contact positions in the scene are not moving (*i.e.* zero velocity). This is a reasonable assumption since most of the contact positions in daily life in a static scene are static contacts, which is also the case with our evaluation dataset; GPA dataset (Wang et al., 2022b, 2020b).

With the sliding loss term, our framework reduces the sliding error by ∼14% compared to w/o $\mathcal{L}_{\text{sli}}$. Notably, integration of $\mathcal{L}_{\text{sli}}$ reduces the 3D joint error (MPJPE) by 1% as well.

Contact Classifications As HULC is the first method estimating contact labels on dense body and environment surfaces from monocular RGB and point cloud input, there are no other existing works that estimate the same outputs. Nonetheless, we report the performance on the GPA dataset for completeness and future reference. The precision, recall and accuracy of the body surface contact estimation are 0.22, 0.41 and 0.91, respectively.

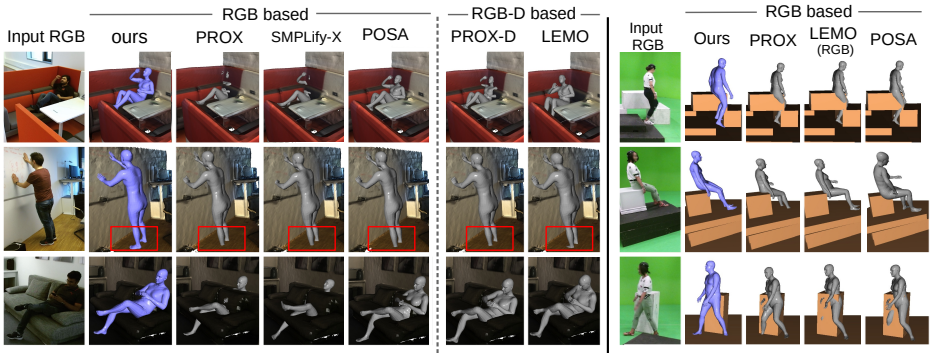


Figure 5.5: The qualitative comparisons of our results with the related methods on PROX (left) and GPA dataset (right). Our RGB-based HULC shows fewer body-scene penetrations even when compared with RGB-D based methods; mind the red rectangles in the second row.

For the environment surface contact estimation, 0.045, 0.18 and 0.96, respectively. Note that these classification tasks are highly challenging, especially since the environment point cloud contains several thousands of vertices to be classified. Furthermore, GPA dataset sequences are not included in the training dataset for the contact estimation networks. Although it is conceivable that the reported numbers can be further improved, our framework largely benefits from the estimated contact labels and significantly reduces the 3D localisation errors.

5.5.3 *Qualitative Results*

Fig. 5.5 summarises the qualitative comparisons on GPA and PROX datasets. HULC produces more physically plausible global 3D poses with mitigated collisions, whereas the other methods show body-environment penetrations. Even compared with the RGB(D) approaches, HULC mitigates collisions (mind the red rectangles).

5.6 CONCLUDING REMARKS

Limitations HULC requires the scene geometry aligned in a camera frame like other related works (Hassan et al., 2019, 2021b; Zhang et al., 2021d). Also, HULC does not capture non-rigid deformations of scenes and bodies, although the body surface and some objects in the

environment deform (*e.g.* when sitting on a couch or lying in a bed). Moreover, since our algorithm relies on the initial root-relative pose obtained from an RGB-based MoCap algorithm, the subsequent steps can fail under severe occlusions. Although the estimated contact labels help to significantly reduce the 3D translation error, the estimated environment contacts contain observable false positives.

Conclusion We introduced *HULC*—the first RGB-based scene-aware MoCap algorithm that estimates and is guided by dense body-environment surface contact labels combined with a pose manifold sampling. HULC shows 60% smaller 3D-localisation errors compared to the previous methods. Furthermore, deep body-environment collisions are handled as a hard constraint in the pose manifold sampling-based optimisation, which significantly mitigates collisions with the scene. HULC shows the lowest number of collisions even compared with RGBD-based scene-aware methods.

6

DECAF: MONOCULAR DEFORMATION CAPTURE FOR FACE AND HAND INTERACTIONS

The previous chapter introduced a new scene-aware motion capture approach. This approach leverages the estimated whole-body contacts for improved 3D accuracy while resolving the collisions with the novel sampling optimisation step in a pose manifold space. While the proposed method shows improved performances over the prior works in terms of 3D accuracy and interaction plausibility, the interacting scene is assumed to be static. However, in the real world, daily interactions often result in observable non-rigid effects.

This chapter (published as Shimada et al., 2023) presents the first technique that predicts the hand and face motions, along with the non-rigid skin deformations resulting from their interactions, all from a single-view RGB video. Due to the lack of a suitable training dataset for this problem, this chapter proposes a new 3D deformation dataset with corresponding multi-view RGB images. It is generated using a marker-less multi-view capture system combined with a deformable object simulator. For the simulation, this chapter also proposes a novel non-uniform stiffness computation that considers the underlying skull geometry of a human head. The effectiveness of the locally varying stiffness values is demonstrated qualitatively. The networks trained on the new dataset regress plausible deformations and contacts only from RGB inputs, which are subsequently leveraged in the final global fitting optimisation step. To address the inherent depth ambiguity of the single-view setup, a novel variational autoencoder-based interaction prior is also integrated into the global fitting optimisation. The ablation study validates the importance of this component. The reconstructed motions from the proposed method show the lowest 3D errors for the hand and face motions compared with other related works while being the first to model the deformations arising from interactions.

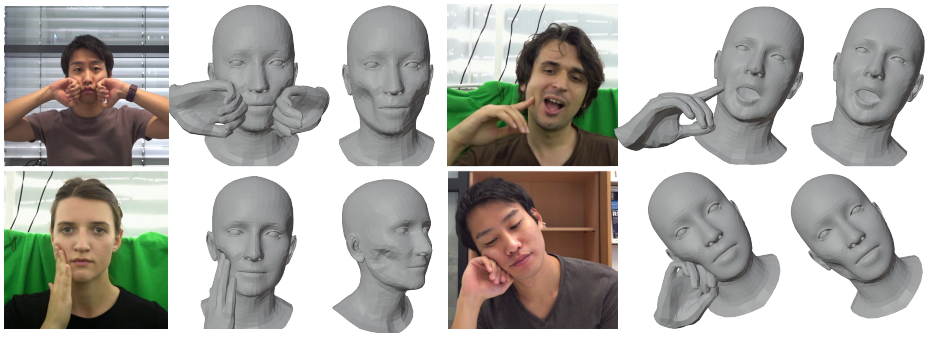


Figure 6.1: Our *Decaf* approach captures hands and face motions as well as the *face surface deformations* arising from the interactions from a single-view RGB video.

6.1 INTRODUCTION

The reconstruction of 3D hands and face models from a monocular RGB video is a challenging and important research area in computer graphics. The task becomes significantly more difficult when attempting to reconstruct hands and face simultaneously including *surface deformations caused by their interactions*. Capturing such interactions and deformations is crucial for enhancing realism in reconstructions as they are frequently observed in everyday life (hand-face interaction occurs 23 times per hour on average during awake-time (Kwok et al., 2015)), and they significantly impact the impressions formed by others. Consequently, reconstructing hand-face interactions is key for avatar simulation, virtual/augmented reality, character animation, where realistic facial movements are essential to create an immersive experience, as well as for applications such as sign language transcriptions and driver drowsiness monitoring. Despite several studies on the reconstruction of face and hand motions, the capture of interactions between them and the corresponding deformations from a monocular RGB video remains unaddressed (Tretschk et al., 2023). On the other hand, naïvely using existing template-based hand and face reconstruction methods leads to artefacts such as collisions, and missing interactions and deformations due to the inherent depth ambiguity in the monocular setting and the lack of deformation modelling in the reconstruction pipeline.

Several key challenges are associated with this problem setting. One (I) is the lack of an available markerless RGB capture dataset for face and

hand interaction with non-rigid deformations for model training and method evaluation. Capturing such a dataset is highly challenging due to the constant presence of occlusions caused by hand and head motions, particularly at the interaction region where non-rigid deformation occurs. Another challenge (II) is the inherent depth ambiguity of the single-view RGB setup, which makes it difficult to obtain accurate localisation information, resulting in errors that can cause artefacts such as collisions or non-touching of the hand and head (when they interact in practice). To tackle these challenges, we propose *Decaf* (short for *deformation capture of faces interacting with hands*), a monocular RGB method for capturing face and hand interactions along with facial deformations.

Specifically, to address (I), we propose a solution that combines a multiview capture setup with a position-based dynamics simulator for reconstructing the interacting surface geometry, even under occlusions. To integrate the deformable object simulator, we calculate the stiffness values of a head mesh using a simple but effective "skull-skin distance" (SSD) method. This approach provides non-uniform stiffness to the mesh, which significantly improves the qualitative plausibility of the reconstructed geometry compared with uniform stiffness values. To address the challenge (II), we train the networks to obtain the 3D surface deformations, contact regions on the head and hand surfaces, and the interaction depth prior from single-view RGB images utilising our new dataset. During the final optimisation stage, we utilise these information from different modalities to obtain plausible 3D hand and face interactions with non-rigid surface deformations, which helps disambiguate the depth ambiguity of the single-view setup. Our approach results in much more plausible hands-face interactions compared to the existing works; see Fig. 6.1 for representative results.

In summary, the primary technical contributions of this chapter are as follows:

- *Decaf* , the first MoCap approach for 3D hand and face interaction reconstruction with face surface deformations (Sec. 6.3).
- A global fitting optimisation guided by the estimated contacts, learned interaction depth prior, and deformation model of the face to enable plausible 3D interactions (Sec. 6.3.3).

- The acquisition of the first markerless RGB-based 3D hand-face interaction dataset with surface deformations with consistent topology based on position-based dynamics (PBD). The reference 3D data for model training and evaluation are generated using a simple and effective non-uniform stiffness estimation approach for human head models, namely *skull-skin distance* (*SSD*; Sec. 6.4).

Our *Decaf* outperforms benchmark and existing related methods both qualitatively and quantitatively, with notable improvements in physical plausibility metrics (Sec. 6.5.3).

6.2 RELATED WORK

This section focuses on the 3D reconstruction of hands interacting with objects in the monocular (single-view) capture context.

6.2.1 *Hand Reconstruction with Interactions*

There have been diverse works proposed to capture 3D hand motions with interactions. Several works reconstruct 3D hand and rigid object interactions from depth information (Hu et al., 2022; Zhang et al., 2019, 2021b) or RGB camera (Cao et al., 2021; Grady et al., 2021; Liu et al., 2021; Tekin et al., 2019). There are several works that reconstruct hand-hand interactions. Mueller et al. (2019) reconstruct two hands interactions from a single depth camera utilising collision proxies based on Gaussian spheres embedded in the hand model. Some works reconstruct interacting 3D hands from a single RGB image (Wang et al., 2022a; Zhang et al., 2021a). However, none of these works considers the non-rigidity while interactions unlike ours.

Similar to our approach, Tsoli and Argyros (2018) reconstruct **non-rigid** cloth and hand interaction by considering hand/object contact points in the optimisation. However, the method requires **RGB-D** input unlike ours. Our work assumes no access to depth sensor information and reconstructs interactions with a deformable face. The face exhibits varying stiffness values based on the surface area, owing to the underlying skull structure in a human's head. This is in contrast to cloth interactions, which typically have uniform stiffness values. Furthermore, our face

autonomously changes its pose and expression during the sequence, whereas in Tsoli and Argyros (2018), the behaviour of the cloth changes only due to the interacting hand or gravity. These unique characteristics, coupled with the limited input setting, make our problem highly challenging.

6.2.2 *Monocular Face Reconstruction*

Capturing a human face from a single view RGB input is important for many graphics applications, thus a significant amount of works have been proposed with learning-free (Garrido et al., 2013, 2016; Thies et al., 2016; Wu et al., 2016) and learning-based approaches (Ichim et al., 2015; Lattas et al., 2020; Saito et al., 2016). In this category, some works train the networks in a self-supervised manner to reconstruct faces with textures and illuminations (Tewari et al., 2017) or details with estimated normals (Danecek et al., 2022; Feng et al., 2021b). Although these works capture the geometry of expressive deforming human faces, none of the works in this category models the face deformations caused by the interactions unlike ours.

6.2.3 *Shape from Template (SfT)*

This algorithm class bears a similarity to our approach. SfT assumes a template mesh of the tracking object and deforms the template mesh based on the observations such as RGB/-D sequences. Several works address this problem with learning-based algorithms (Bozic et al., 2020; Fuentes-Jimenez et al., 2021; Golyanik et al., 2018; Kairanda et al., 2022; Shimada et al., 2019), and some with learning-free optimisation-based approaches (Habermann et al., 2018; Ngo et al., 2015; Salzmann et al., 2007; Yu et al., 2015; Zollhöfer et al., 2014). Unlike these approaches, our method models *interactions* between two different objects (*i.e.* hand and face) from a single view RGB input under severe occlusions caused by the interactions. Petit et al. (2018) propose a physics-based non-rigid object tracking method using a finite element method. However, their method requires RGB-D input and focuses on simple deformable objects (*e.g.*, cubes and discs). In contrast, our approach does not rely on depth infor-

mation and handles interactions between a complex articulated hand and face, considering locally varying stiffness values. Some works estimate 3D human poses with self- and multi-person interactions (contacts) from single RGB images (Fieraru et al., 2020, 2021; Müller et al., 2021). However, they do not model significant surface deformations due to contacts (*e.g.* during hand-face interactions). Li et al. (2022) propose a method that addresses a problem set that bears resemblance to ours. It estimates the 3D global human pose along with the deformations of the interacting environment surface based on ARAP-loss. However, their method does not consider stiffness values specific to object categories and does not incorporate learned priors for non-rigid deformations, distinguishing it from our approach.

6.2.4 *Template Free Non-Rigid Surface Tracking*

Some methods in this category reconstruct non-rigid surfaces by acquiring first an explicit template mesh from RGB-D inputs (Innmann et al., 2016). Some use node graphs (Lin et al., 2022) or implicit SDF surface representations (Slavcheva et al., 2017) for non-rigid surface tracking. Guo et al. (2017) propose a method that reconstructs the non-rigid surface along with the surface albedo and low-frequency lighting. Our approach differs from these works by considering the dynamics of the interactions between two different materials *i.e.* face and hand, and face surface stiffness values based on bone structure. Additionally, our dataset and method's output have consistent 3D mesh topologies that are very important for the supervision of network training in explicit surface space.

6.2.5 *Physics-based MoCap*

Recently, numerous physics-based algorithms for motion capture have been proposed. Several works model the interactions with the environment from a static single RGB camera (Gärtner et al., 2022a,b; Huang et al., 2022; Innmann et al., 2016; Luo et al., 2022; Rempe et al., 2020; Shimada et al., 2021, 2020; Xie et al., 2021; Yuan et al., 2021) or with objects (Dabral et al., 2021). Some works reconstruct 3D poses from egocentric views (Luo et al., 2021) or IMUs (Yi et al., 2022). Hu et al. (2022)

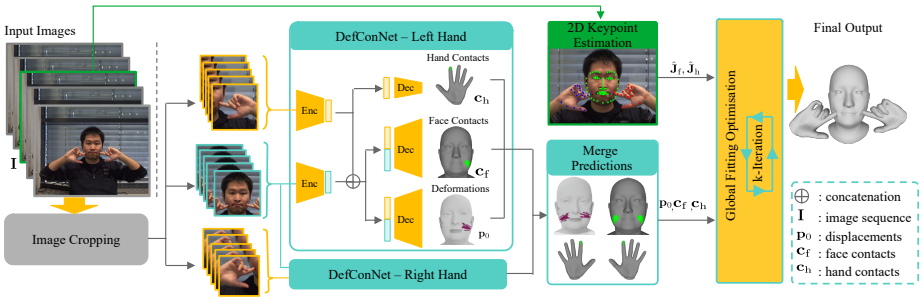


Figure 6.2: Schematic visualisation of *Decaf* , the proposed system to predict 3D poses of hands and face in interaction from a sequence of monocular RGB images of a subject. The final output from *Decaf* reconstructs the face and hands, incorporating plausible surface deformations on the face resulting from their interactions.

reconstruct hand-object interactions from an RGB-D camera sequence modelling the physics-based contact status. While the existing approaches primarily focus on modelling the interactions with static floor planes or rigid objects, our method uniquely addresses non-rigid deformations arising from interactions between hands and face. This capability is made possible thanks to our networks trained on our novel dataset, which incorporates 3D deformations generated using a maker-less multiview motion capture system combined with position based dynamics (PBD) (Müller et al., 2007) – a widely adopted deformable object simulation algorithm employed in modern physics engines.

6.3 METHOD

Our goal is to reconstruct hands interacting with a face in 3D, including non-rigid face deformations caused by the interaction, from a single monocular RGB video. Fig. 6.2 provides an overview of the proposed framework. Our deformation and contact estimation network *DefConNet*, trained on our new dataset (Sec. 6.4), estimates face surface deformations and contact labels on both face and hand surfaces from an image sequence; the contact labels are crucial to achieve plausible and realistic interactions in 3D (Sec. 6.3.2). The estimated deformations, contacts and 2D keypoints are subsequently sent to the global fitting optimisation stage (Sec. 6.3.3), where we also utilise the

interaction prior obtained from a conditional variational autoencoder (Sohn et al., 2015) conditioned on the 2D key points for the improved interactions between the hands and face. After this stage, we obtain the final 3D reconstruction of the face and hands in the form of parametric hand and head models with applied deformations. We next explain the notations and assumptions that are used in this work (Sec. 6.3.1), followed by the details of our *Decaf* approach.

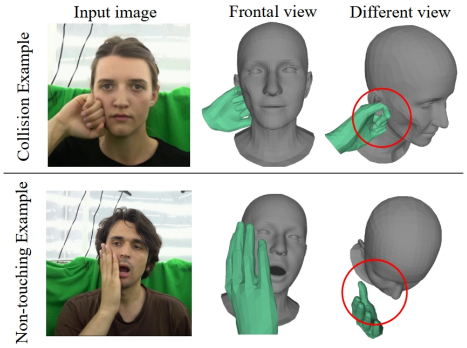


Figure 6.3: Example artefacts caused by the depth inaccuracies after solving a naïve single RGB based fitting optimisation, *i.e.* Eqs. (6.5) and (6.8) without $\mathcal{L}_{\text{touch}}$, $\mathcal{L}_{\text{col.}}$ and $\mathcal{L}_{\text{depth}}$. The locations of the observable artefacts are indicated by the red circles on each row.

6.3.1 *Modelling and Preliminaries*

Our *Decaf* accepts as input a sequence $\mathbf{I} = \{\mathbf{I}_t\} = \{\mathbf{I}_1, ..., \mathbf{I}_T\}$ of $T = 5$ successive RGB frames from a static camera with known intrinsic camera parameters. We resize $\mathbf{I}_t$ to 224×224 pixels after cropping the detected bounding box around the subject's face and hands in each frame. To represent the 3D face, we employ a gender-neutral version of FLAME parametric model $\mathcal{F}$ (Li et al., 2017). We utilise its identity parameters $\boldsymbol{\beta}_\text{f} \in \mathbb{R}^{100}$, jaw pose $\boldsymbol{\theta}_\text{f} \in \mathbb{R}^3$ and expression parameters $\boldsymbol{\Psi} \in \mathbb{R}^{50}$ combined with the global translation $\boldsymbol{\tau}_\text{f} \in \mathbb{R}^3$ and rotation $\mathbf{r}_\text{f} \in \mathbb{R}^3$ that can be formulated as a differentiable function $\mathcal{F}(\boldsymbol{\tau}_\text{f}, \mathbf{r}_\text{f}, \boldsymbol{\beta}_\text{f}, \boldsymbol{\theta}_\text{f}, \boldsymbol{\Psi})$. Model $\mathcal{F}$ returns 3D head vertices $\mathbf{V}_\text{f} \in \mathbb{R}^{M\times 3}$ (M = 5023) from which we obtain the 3D face landmarks $\mathbf{J}_\text{f} \in \mathbb{R}^{K_\text{f}\times 3}$ (K_f = 68). To represent 3D hands, we employ the gender neutral version of the statistical MANO parametric hand model (Romero et al., 2017) that defines the hand mesh as a function $\mathcal{M}(\boldsymbol{\tau}_\text{h}, \mathbf{r}_\text{h}, \boldsymbol{\theta}_\text{h}, \boldsymbol{\beta}_\text{h})$ of global translation $\boldsymbol{\tau}_\text{h} \in \mathbb{R}^3$ and global root orientation $\mathbf{r}_\text{h} \in \mathbb{R}^3$, pose parameters $\boldsymbol{\theta}_\text{h} \in \mathbb{R}^{45}$ and hand identity parameters $\boldsymbol{\beta}_\text{h} \in \mathbb{R}^{10}$. This function $\mathcal{M}$ returns hand 3D mesh vertices $\mathbf{V}_\text{h} \in \mathbb{R}^{N\times 3}$ (N = 778) from which 3D hand joint positions $\mathbf{J}_\text{h} \in \mathbb{R}^{K_\text{h}\times 3}$ (K_h = 21) are obtained. We assume that the face identity and hand shape parameters are known. In the following, $\boldsymbol{\Phi}_\text{f} = (\boldsymbol{\tau}_\text{f}, \mathbf{r}_\text{f}, \boldsymbol{\beta}_\text{f}, \boldsymbol{\theta}_\text{f}, \boldsymbol{\Psi})$ and $\boldsymbol{\Phi}_\text{h} = (\boldsymbol{\tau}_\text{h}, \mathbf{r}_\text{h}, \boldsymbol{\beta}_\text{h}, \boldsymbol{\theta}_\text{h})$ denote the kinematic states of the face and hand in a 3D space.

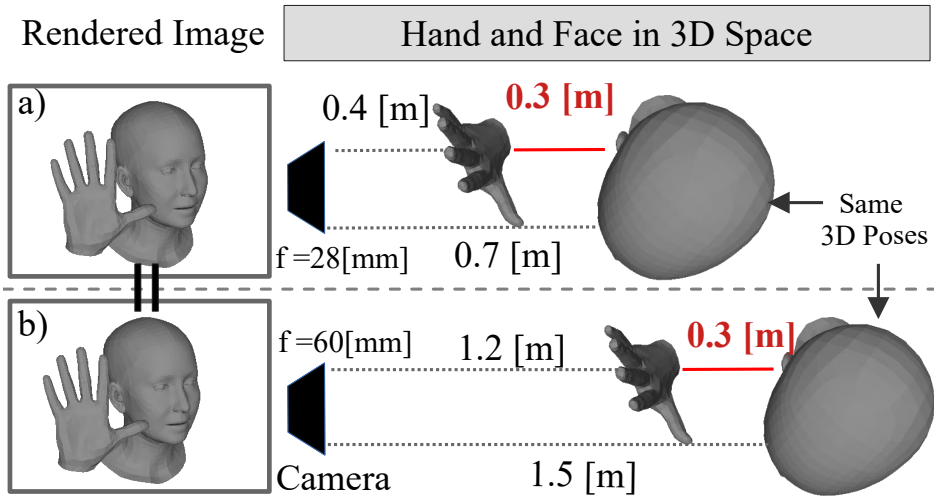


Figure 6.4: Schematic visualisation of depth ambiguity in a monocular setup. f denotes the focal length of the camera. **a) and b):** Given the same 3D poses of face and hand of the same scale in the 3D space, different combinations of depths and focal lengths can result in indistinguishable images after the 2D projection in a monocular setting. This effect, known as depth ambiguity, poses a challenge for methods attempting to estimate the depth values of the hand and face in the camera frame from monocular 2D inputs (*e.g.* RGB images or 2D keypoints). However, the relative location of the hand w.r.t. the head is invariant to the positions of the face and hand in 3D space (*e.g.* 0.3 [m] above). Based on this idea, our DePriNet learns the depth prior in the *canonical face frame* where the origin of the frame is located at the centre of the head.

6.3.2 *Interaction Estimation*

We introduce a learning-based approach that estimates plausible interactions in a scene, *i.e.* the vertex-wise face deformations and contacts on the face and hand surfaces given only single-view RGB images. The approach is trained on our new dataset (Sec. 6.4).

Our neural network accepts as input an image sequence $\mathbf{I}$ and outputs the deformation on the head model as per-vertex displacements in a camera frame $\mathbf{p} \in \mathbb{R}^{M\times 3}$, contact labels on the face $\mathbf{c}_f \in \{0,1\}^M$ and the hand $\mathbf{c}_h \in \{0,1\}^N$. The contact labels are binary signals, *i.e.* 1 for contact, 0 otherwise. The network is trained to estimate the contact probability using the binary cross entropy (BCE):

$$\mathcal{L}_{\text{labels}} = \text{BCE}(\mathbf{c}_f, \hat{\mathbf{c}}_f) + \text{BCE}(\mathbf{c}_h, \hat{\mathbf{c}}_h), \tag{6.1}$$

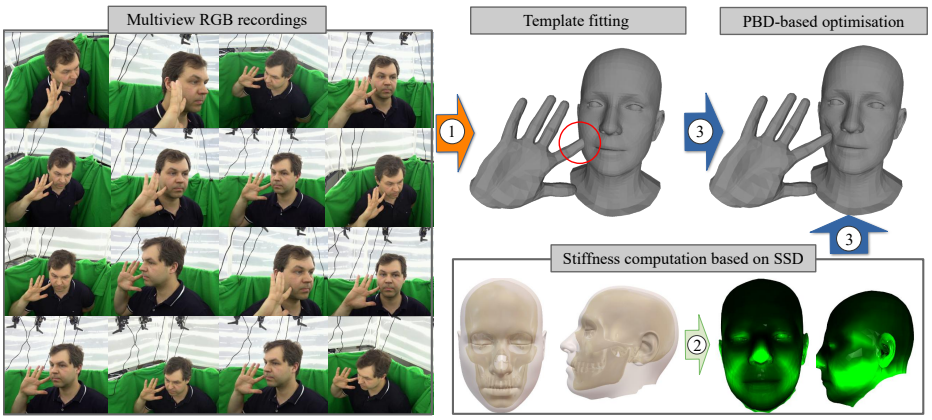


Figure 6.5: Overview of the dataset generation pipeline. We first capture the hand and face interactions using a markerless multi-view setup. **(1)** Subsequently, the obtained RGB image sequences are used to solve template-based fitting optimisation. **(2)** To provide the plausible stiffness values on the head mesh for the later position-based dynamics (PBD) optimisation stage, we compute skull-skin distances (SSD) and obtain vertex-wise stiffness values, see Sec. 6.4.2 for the details. **(3)** Using the fitted templates from (1) and the stiffness values from (2), we solve the PBD-based tracking optimisation. This stage handles the physically implausible collisions and provides plausible surface deformations on the head mesh surface (Sec. 6.4.3).

where $\hat{\mathbf{c}}_\text{f}$ and $\hat{\mathbf{c}}_\text{h}$ denote the ground-truth contact labels for the face and hand, respectively. We also train the network to estimate the deformations using the ground-truth annotations $\hat{\mathbf{p}}_m$:

$$\mathcal{L}_\text{def.} = \frac{1}{M}\sum_{m=1}^{M}(w_\text{def}^m \left\|\mathbf{p}_m - \hat{\mathbf{p}}_m\right\|_2^2 + b_\text{def}^m \left\|\mathbf{p}_m\right\|), \tag{6.2}$$

where

$$w_\text{def}^m = \begin{cases} 0.3, & \text{if} \quad ||\hat{\mathbf{p}}_m|| = 0, \\ 1.0, & \text{otherwise}, \end{cases} \qquad b_\text{def}^m = \begin{cases} 1, & \text{if} \quad ||\mathbf{p}_m|| > \psi, \\ 0, & \text{otherwise}. \end{cases} \tag{6.3}$$

The first term in Eq. (6.2) allows the network to learn the 3D deformations in our dataset. The weight w_def helps to penalise the network predictions more on deforming vertices. We observe that this weighting strategy improves the network precision as the majority of the face vertices have no deformations. The second loss term in Eq. (6.2) regularises the unnaturally large deformations on the face surface where b_def works as a

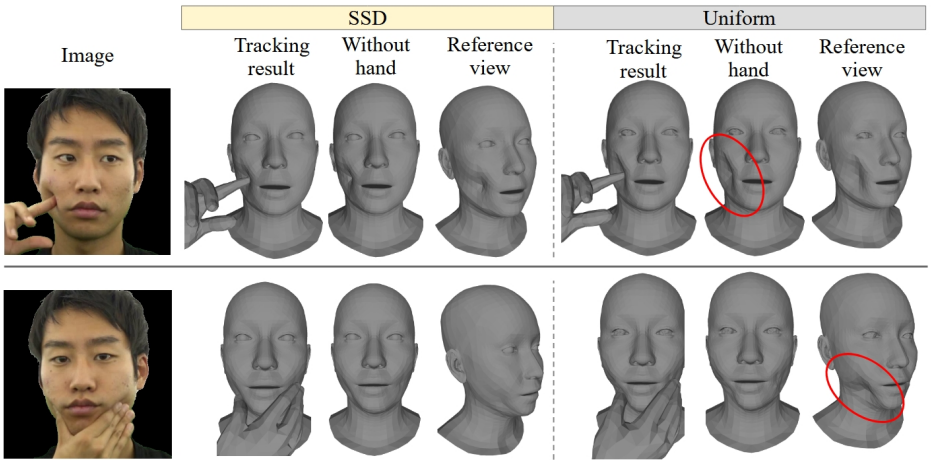


Figure 6.6: Example visualisations of the reconstructed 3D head and hand interactions with the stiffness values computed using the skull-skin distance (SSD) (second to fourth columns) and the uniform stiffness value (fifth to seventh columns). With SSD, the obtained surface deformations are much more plausible compared to naïevely assigning the uniform stiffness value to all the head vertices. The red circles highlight the overly deformed surfaces (top) and inaccurate deformations that ignore the underlying jaw in the human head (bottom).

binary label to penalise only the vertices with deformations greater than $\psi = 0.1$ [m].

6.3.3 *Global Fitting Optimisation*

Using the estimated deformations $\mathbf{p}$, contact labels $\mathbf{c}_f$ and $\mathbf{c}_h$ and 2D joint keypoints, we obtain the global positions of the face $\mathbf{\Phi}_f$ and hand $\mathbf{\Phi}_h$ in the 3D scene considering their interactions. In this optimisation step, we also update $\mathbf{p}$ to refine and handle the minor collisions. The objective follows:

$$\mathcal{L}_{\text{opt}}(\mathbf{\Phi}_f, \mathbf{\Phi}_h, \mathbf{p}) = \mathcal{L}_{\text{face}} + \mathcal{L}_{\text{hand}}. \tag{6.4}$$

The fitting loss term of the face model $\mathcal{L}_{\text{face}}$ reads:

$$\mathcal{L}_{\text{face}}(\mathbf{\Phi}_f, \mathbf{p}) = \mathcal{L}_{2D} + \mathcal{L}_{\text{reg.}}, \tag{6.5}$$

where $\mathcal{L}_{2D}$ and $\mathcal{L}_{\text{reg.}}$ are the weights of the 2D reprojection term and regulariser loss term , respectively. Employing the projection function

$\Pi(\cdot)$ with the known camera intrinsics, the 2D reprojection loss term is formulated as follows:

$$\mathcal{L}_{\text{2D}} = \frac{1}{M}\sum_{m=1}^{M} w_{\text{conf.}}^{m} \|\Pi(\mathbf{J}_{\text{f}}^{m}) - \hat{\mathbf{J}}_{\text{f}}^{m}\|_2^2, \tag{6.6}$$

where $\hat{\mathbf{J}}_{\text{f}}^{m}$ and $w_{\text{conf.}}^{m}$ are, respectively, the reference 2D face landmarks and the corresponding confidence value obtained by the method of Bulat and Tzimiropoulos (2017) given the input image. We also minimise the regulariser loss term $\mathcal{L}_{\text{reg.}}$ to introduce the statistical prior for the shape $\boldsymbol{\beta}_{\text{f}}$ and expression $\boldsymbol{\Psi}$, and temporal smoothness in the motion:

$$\mathcal{L}_{\text{reg.}} = \lambda_{\boldsymbol{\beta}}\|\boldsymbol{\beta}_{\text{f}}\|_2^2 + \lambda_{\boldsymbol{\Psi}}\|\boldsymbol{\Psi}\|_2^2 + \lambda_{\dot{\mathbf{V}}}\|\dot{\mathbf{V}}_{\text{f}}\|_2^2 + \lambda_{\ddot{\mathbf{V}}}\|\ddot{\mathbf{V}}_{\text{f}}\|_2^2, \tag{6.7}$$

where $\dot{\mathbf{V}}_{\text{f}}$ and $\ddot{\mathbf{V}}_{\text{f}}$ denote the velocity and acceleration of the head vertex positions $\mathbf{V}_{\text{f}}$, respectively. $\lambda_{\bullet}$ denotes a weight of the loss term. The objective for the hand fitting $\mathcal{L}_{\text{hand}}$ optimisation includes the 2D reprojection term $\mathcal{L}_{\text{2D}}$, regulariser term $\mathcal{L}_{\text{reg.}}$, collision term $\mathcal{L}_{\text{col.}}$, *touchness* term $\mathcal{L}_{\text{touch}}$ and the depth prior term $\mathcal{L}_{\text{depth}}$:

$$\begin{aligned}\mathcal{L}_{\text{hand}}(\boldsymbol{\Phi}_{\text{h}}, \mathbf{p}) = \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{reg.}} + \lambda_{\text{touch}}\mathcal{L}_{\text{touch}} \\ + \lambda_{\text{col.}}\mathcal{L}_{\text{col.}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}},\end{aligned} \tag{6.8}$$

where $\lambda_{\bullet}$ are the corresponding weights. The terms $\mathcal{L}_{\text{2D}}$ and $\mathcal{L}_{\text{reg.}}$ are the same as in (6.6)-(6.7) with the modification that (6.6) is applied on the hand 3D joints $\mathbf{J}_{\text{h}}$ compared with the reference 2D hand keypoints $\hat{\mathbf{J}}_{\text{h}}$, and (6.7) on the hand shape $\boldsymbol{\beta}_{\text{h}}$, velocity and acceleration of hand vertices, excluding the expression prior loss $\|\boldsymbol{\Psi}\|_2^2$.

Due to the inaccuracy of the depth estimation in the monocular setting, simply solving the fitting optimisation w.r.t. the face and hand global positions can cause artefacts, *e.g.* collisions between the face and hand or non-touching artefacts. Fig. 6.3 shows examples of such artefacts, when solving a naïve 2D reprojection based single view fitting optimisation *i.e.* (6.4) excluding $\mathcal{L}_{\text{touch}}$, $\mathcal{L}_{\text{col.}}$ and $\mathcal{L}_{\text{depth}}$. They immediately give the impression of unnatural hand-face interaction to the viewer. To address the "non-touching" artefacts, we utilise the *touching* loss term $\mathcal{L}_{\text{touch}}$ that penalises the distances between the contact surfaces on the face and hands inspired by Shimada et al. (2022). Specifically, we treat the face and

hand vertices with contact probabilities $\mathbf{c}_\mathrm{f} > 0.5$ and $\mathbf{c}_\mathrm{h} > 0.5$ as effective contacts, respectively. Let $\mathcal{C}_\mathrm{f} \subset [\![1, n]\!]$ and $\mathcal{C}_\mathrm{h} \subset [\![1, m]\!]$ be the index subsets of the face and hand vertices with the effective contacts. Using a Chamfer loss, $\mathcal{L}_\mathrm{touch}$ is formulated as follows:

$$\mathcal{L}_\mathrm{touch} = \frac{1}{|\mathcal{C}_\mathrm{f}|} \sum_{i \in \mathcal{C}_\mathrm{f}} \min_{j \in \mathcal{C}_\mathrm{h}} \left\| \mathbf{V}_\mathrm{f}^i - \mathbf{V}_\mathrm{h}^j \right\|_2^2 + \frac{1}{|\mathcal{C}_\mathrm{h}|} \sum_{j \in \mathcal{C}_\mathrm{h}} \min_{i \in \mathcal{C}_\mathrm{f}} \left\| \mathbf{V}_\mathrm{f}^i - \mathbf{V}_\mathrm{h}^j \right\|_2^2. \tag{6.9}$$

To avoid collisions between hands and a head, we also introduce the collision loss term $\mathcal{L}_\mathrm{col.}$ for minimising the penetration distance of the hand vertices. Specifically, we first detect the hand vertices colliding with the face mesh based on an SDF criterion (Yu, 2023). Then, we minimise the distance between colliding hand vertices and their nearest vertices on the head mesh. Let $\mathcal{P} \subset [\![1, W]\!]$ be the subset of indices of hand vertices $\mathbf{V}_\mathrm{h}$ colliding with the face mesh. The collision loss is formulated as:

$$\mathcal{L}_\mathrm{col.} = \sum_{i \in \mathcal{P}} \min_{j \in \mathcal{V}_\mathrm{f}} \left\| \mathbf{V}_\mathrm{h}^i - \mathbf{V}_\mathrm{f}^j \right\|_2^2 + \mathcal{L}_\mathrm{regDef}, \tag{6.10}$$

where $\mathcal{V}_\mathrm{f} \subset [\![1, M]\!]$ is the set of all the indices of the face vertices $\mathbf{V}_\mathrm{f}$. The term $\mathcal{L}_\mathrm{regDef}$ regularises the update of the deformation $\mathbf{p}$ from the perspective of edge lengths, neighbouring face angles and original deformation estimated by DefConNets. Let $l = \{l_1, ..., l_x\}$ and $\varphi = \{\varphi_1, ..., \varphi_y\}$ be vectors that consist of the edge lengths and the angles between the neighbouring faces of the face mesh, respectively. The formulation of $\mathcal{L}_\mathrm{regDef}$ reads:

$$\mathcal{L}_\mathrm{regDef} = \sum_{i=1}^{x} s_\mathrm{edge}^i \left\| l_i - l_0 \right\|_2^2 + \sum_{i=1}^{y} s_\mathrm{bend}^i \left\| \varphi_i - \varphi_0 \right\|_2^2 + \left\| \mathbf{p} - \mathbf{p}_0 \right\|_2^2, \tag{6.11}$$

where l_0 and φ_0 denote the edge lengths and dihedral angles at rest and $\mathbf{p}_0$ is the displacements estimated by DefConNets in the previous step; s_edge and s_bend are, respectively, the edge and bending stiffness values that consider the underlying skull structure of a human head. The details of the stiffness computations are elaborated in Sec. 6.4.2.

To further introduce the learned prior for the depth position of the hand, we train a conditional variational autoencoder (CVAE) (Sohn et al., 2015) -based depth prior network *DePriNet* that is conditioned on the 2D key points. DePriNet is trained to reconstruct the 3D hand key points

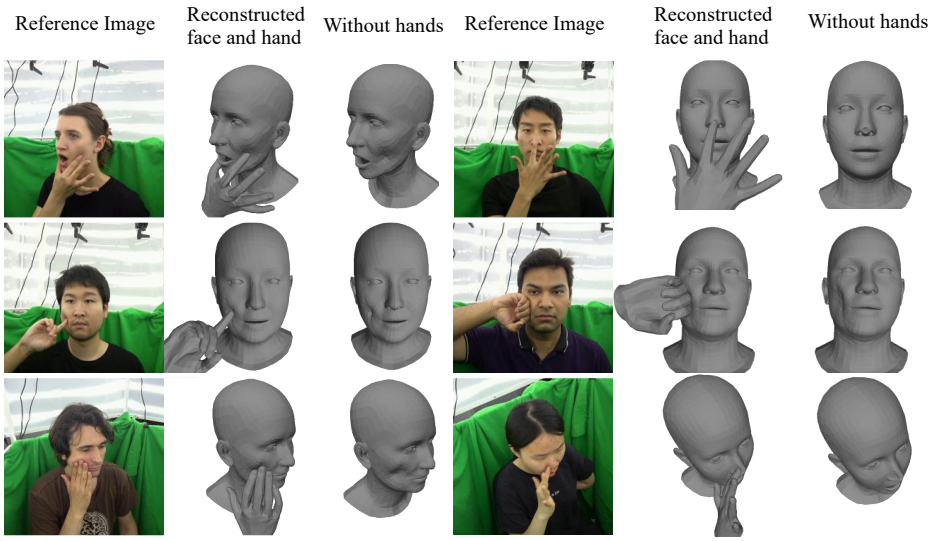


Figure 6.7: Example visualisations from our new hands+face 3D motion capture dataset with hand shape articulations non-rigid face deformation. The reconstructed 3D geometry shows plausible surface deformations thanks to the fitting optimisation combined with PBD.

in a **canonical face frame**, as estimating the depth of hand and face in the camera frame only from monocular 2D input is challenging due to the depth ambiguity (*e.g.* 3D hand and face with different combinations of focal lengths and depths can be projected onto the same position in the 2D image). However, the hand positions relative to the face in the 3D space are invariant to the depth in the camera frame; see Fig. 6.4 for a schematic visualisation. We train DePriNet with the standard losses:

$$\mathcal{L}_{\text{vae}} = \left\| \mathbf{J}_{\text{h}}^{*} - \hat{\mathbf{J}}_{\text{h}}^{*} \right\|_2^2 + \text{KL}\left(q(\mathbf{Z} \mid \hat{\mathbf{J}}_{\text{h}}^{*}, \boldsymbol{\Theta}) \| \mathcal{N}(\mathbf{0}, \mathbf{I}) \right). \tag{6.12}$$

The first term is a reconstruction loss to reproduce the ground-truth input hand joints in a canonical face frame $\hat{\mathbf{J}}_{\text{h}}^{*} \in \mathbb{R}^{K_{\text{h}} \times 3}$ and $\mathbf{J}_{\text{h}}^{*} \in \mathbb{R}^{K_{\text{h}} \times 3}$ denotes the output from the decoder network of DePriNet. The second loss term penalises the deviation of the latent vector $\mathbf{Z} \in \mathbb{R}^{50}$ distribution from a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ using the Kullback-Leibler divergence loss $\text{KL}(\cdot \| \cdot)$. Latent $\mathbf{Z}$ is sampled from a Gaussian distribution whose mean and variance are estimated from the encoder network $q(\cdot)$ of DePriNet. At test time, we use the decoder network $p(\cdot)$ of DePriNet

to output depth candidates of the hand positions that are integrated into the depth prior loss $\mathcal{L}_{\text{depth}}$ in the global fitting optimisation:

$$\mathcal{L}_{\text{depth}} = \sum_{i=1}^{u} w_i \left\| \mathbf{J}_{\text{h}}^{z} - \mathbf{T}(\mathbf{J}_{\text{h},i}^{*}) \right\|_2^2, \tag{6.13}$$

$$\text{where } w_i = 1 - \frac{\eta_i - \min(\eta)}{\max(\eta) - \min(\eta)}, \quad \eta_i = |\mathbf{Z}^i|_1, \tag{6.14}$$

$\mathbf{J}_{\text{h}}^{z}$ denotes the z-value of the hand 3D keypoints $\mathbf{J}_{\text{h}}$ that corresponds to the depth axis in the camera frame, and $\mathbf{T}(\cdot)$ is a transformation from the canonical face space to the camera frame that consists of the rotation and translation of the face model (that are also simultaneously obtained in this global fitting optimisation); $\mathbf{J}_{\text{h},i}^{*}$ is the i-th sample obtained from the decoder $p(\cdot)$ given $u = 100$ latent vectors $\sim\mathcal{N}(\mathbf{0}, \mathbf{I})$ and the conditioning vector $\boldsymbol{\Theta}$ that consists of face and hand 2D keypoints with corresponding confidence values as well as the face 3D rotation in the camera frame in 6D representation (Zhou et al., 2019). Note that 2D key points of the face and hands are translated to be a face-root relative representation for the conditioning. The conditioning 3D head rotation is obtained during the optimisation (6.4). Each generated sample is weighted by the scalar w that has the higher value the closer the corresponding latent vector $\mathbf{Z}$ is to zero (*i.e.* a statistically more likely sample). We utilise the two independent DePriNets of the same architecture for the left and right hands. After minimising the objective that combines all these loss terms, we obtain the final 3D head and hand reconstructions with plausible deformations and interactions. The significance of each loss term is evaluated in Sec. 6.5. The final deformed face vertices $\mathbf{V}_{\text{f}}^{*}$ are obtained by simply adding the updated deformations $\mathbf{p}$ to the face model parameterised by $\boldsymbol{\Phi}_{\text{f}}$, *i.e.* $\mathbf{V}_{\text{f}}^{*} = \mathcal{F}(\boldsymbol{\Phi}_{\text{f}}) + \mathbf{p}$.

6.3.4 *Architectures of Our Networks*

Our *Decaf* comprises several components (Fig. 6.2). We employ Bulat and Tzimiropoulos (2017) and Lugaresi et al. (2019) for 2D keypoint and bounding box estimation of the face and hand, respectively. The *DefConNet* is composed of two encoders and three decoders. The encoders for the cropped face and hand images follow the ResNet-18 architecture (He et al., 2016). The decoders, sharing the same architecture, estimate

per-vertex deformations and contact labels for the face and hand. Each of them includes three fully connected layers with leaky ReLU activation (Maas et al., 2013) and their hidden layer dimensions equal to 1024. We duplicate *DefConNet* for both hands and compute the union of the face deformations and contacts before the final global fitting optimisation. The *DePriNet* is a variational autoencoder (Kingma and Welling, 2014), consisting of three linear fully connected layers with batch normalisations, ReLU activations (Agarap, 2018), a latent dimension of 50 and hidden size of 128 for both encoders and decoders.

6.4 DATASET

In this work, we build a new markerless multi-view dataset for 3D hand-face interactions for method training and evaluation. It contains eight subjects—captured with 15 SONY DSC-RX0 cameras at 50 fps (*i.e.* from 15 different viewpoints)—along with the corresponding reference 3D geometries of a right hand and head, including surface deformations of the head represented as per-vertex displacements. In total, the dataset contains 100K frames, see Table 6.1 for the details. Each actor performs seven different actions with three different facial expressions. For each captured view, the background masks are obtained using Sengupta et al. (2020). The bounding boxes (for the hands and the faces) and 2D key points (for the faces), are obtained using Lugaresi et al. (2019) and Bulat and Tzimiropoulos (2017), respectively.

In the remainder of this section, we elaborate on our dataset generation pipeline; see Fig. 6.5 for the overview. The first step of the pipeline, *i.e.* multiview template fitting, is explained In Sec. 6.4.1. Next, to obtain a reasonable stiffness value that considers the underlying skull structure of a human face, we introduce a simple but effective skull-skin distance (SSD) approach in Sec. 6.4.2. The computed stiffness values are further utilised in the deformable object simulation relying on *position based dynamics (PBD)*, and we obtain the final 3D geometry with plausible interactions arising from hand-face interactions (Sec.6.4.3).

6.4.1 *Multiview Template Fitting*

We first solve the 2D keypoint reprojection-based fitting optimisation to obtain the MANO (Romero et al., 2017) and FLAME model (Li et al., 2017) parameters, so that the hand and face shapes match the multiview 2D keypoints with known intrinsic and extrinsic calibrations. The objective for the face fitting encompasses (6.6) and (6.7). For the hand, we also minimise (6.6) and (6.7) with the modification that (6.6) is applied on the hand 3D joints $\mathbf{J}_{\mathrm{h}}$, and (6.7) is applied on the hand shape $\boldsymbol{\beta}_{\mathrm{h}}$, velocity and acceleration of hand vertices, excluding the expression loss term $\|\boldsymbol{\Psi}\|_2^2$. However, FLAME does not model the surface deformation caused by the interactions, which can result in physically implausible collisions; see the red circle in Fig. 6.5-(1). We address this limitation by integrating into our tracking pipeline a deformable object simulator relying on position-based dynamics (PBD) (Müller et al., 2007). Our approach assumes non-homogeneous stiffness values of the human face, and we describe next how we obtain those.

6.4.2 *Stiffness on a Head Mesh*

Deformable object simulators require known material stiffness. The stiffness of human face tissues is non-uniform, due to the rich mimic musculature and the skull anatomy. Therefore, assuming uniform stiffness in the whole face and head would result in artefacts when running the simulation; see Fig. 6.6 for the examples. We obtain the non-uniform stiffness values based on a simple but effective *skin-skull distance (SSD)* assumption. It is based on the assumption that our face and head region tend to have higher stiffness when the distance between the skin and skull surface is smaller (*e.g.* forehead), and vice versa (*e.g.* cheek). To compute SSD, we employ the mean skull and skin surface of a statistic model from (Achenbach et al., 2018). The obtained tissue stiffness map is upon our expectation and the corresponding pseudo-ground-truth deformations are used in quantitative experiments in Sec. 6.5.

Let $\mathbf{D} = [d_1, ..., d_h] \in \mathbb{R}^h$ be a set of nearest distances between the skin and skull surfaces computed for all the h skin vertices of Achenbach et al. (2018). The stiffness s of the i-th skin vertex is calculated as follows:

$$\mathbf{s}_i = (1 - \hat{d}_i)^b, \tag{6.15}$$

where $\hat{d}$ is the normalised distance:

$$\hat{d}_i = \frac{d_i - \min(\mathbf{D})}{\max(\mathbf{D}) - \min(\mathbf{D})}, \tag{6.16}$$

with the operators $\min(\cdot)$ and $\max(\cdot)$ to compute the minimum and maximum values of the input vector; b is empirically set to 4. After computing the per-point stiffness $\mathbf{s}_i$, we transfer it to the FLAME head model by finding the corresponding vertices based on the nearest neighbour search after fitting the FLAME head model onto the skin surface model of Achenbach et al. (2018). In Fig. 6.5-(2), we show the visualisation of the assigned stiffness values (more saturated green encodes lower stiffness). The assigned values are expected from the anatomical viewpoint (*e.g.* high stiffness around the head region and low stiffness near the tip of the nose and cheeks). The edge and bending stiffness values in (6.11) are obtained by simply computing the average over the s of vertices that forms the edges and triangles.

6.4.3 *PBD-based Optimisation*

Position based dynamics (PBD) (Müller et al., 2007) is a technique for simulating deformable objects, which gained popularity for its robustness and simplicity; it is widely used in game and physics engines. We utilise PBD to resolve implausible head-hand collisions which are challenging to address in a markerless motion capture setup due to constant occlusions at the interaction regions. We utilise stretch constraint C_{stretch}, bending constraint C_{bend} and collision constraint $C_{\text{collision}}$ in the PBD simulator. For each pair of connected vertices $\mathbf{p}_1$ and $\mathbf{p}_2$ in the mesh, C_{stretch} is defined as follows:

$$C_{\text{stretch}}(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 - \mathbf{p}_2| - l_0, \tag{6.17}$$

where l_0 denotes the rest length of the edge between $\mathbf{p}_1$ and $\mathbf{p}_2$. For each pair of adjacent triangles $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_2)$ and $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4)$, the definition of bending constraint C_{bend} reads:

$$C_{\text{bend}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \\ \text{acos}\left(\frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)|} \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)|}\right) - \varphi_0, \tag{6.18}$$

where φ_0 is the rest angle between the two triangles. Collision constraint $C_{\text{collision}}$ can be integrated for each vertex $\mathbf{p}$:

$$C_{\text{collision}}(\mathbf{p}) = \mathbf{n}^T\mathbf{p} - h = 0, \tag{6.19}$$

where $\mathbf{n}$ and h are the normal of the colliding plane and the distance from the plane that $\mathbf{p}$ should maintain. After resolving collisions, we introduce friction as formulated in Müller et al. (2007) with 0.5 for both kinetic and static friction coefficients.

We also additionally introduce constraint C_{track} for tracking the reference 3D motions obtained in Sec. 6.4.1. More specifically, this tracking constraint minimises the Euclidean distance between the vertex of the template mesh $\mathbf{p}$ and its corresponding vertex $\mathbf{p}_{\text{ref}}$ in the reference mesh from the previous multi-view fitting stage:

$$C_{\text{stretch}}(\mathbf{p}, \mathbf{p}_{\text{ref}}) = |\mathbf{p} - \mathbf{p}_{\text{ref}}|. \tag{6.20}$$

For the simulation, we use the stiffness values obtained in Sec.6.4.2, and finally obtain the 3D geometry of the interacting hand and face with the surface deformations (also see Fig. 6.5-(3) for the example reconstruction).

6.5 EVALUATIONS

We next evaluate our *Decaf* on our new dataset. As there are no existing methods that address the same problem we tackle, we compare our method to a most closely related approach, *i.e.* a monocular full-body capture PIXIE (Feng et al., 2021a) and its variants that reconstruct only hands and face independently, denoted as PIXIE (hand+face). We also compare to our benchmark method that includes hand-only (Lugaresi et al., 2019) and face-only (Li et al., 2017) trackers.

Table 6.1: Details of our new dataset. This dataset contains several types of data including pseudo ground truth of 3D surface deformations represented as 3D displacement vectors for seven different actions with three different facial expressions performed by eight subjects. The "Age" signifies the age range, whereas the number in the brackets means the corresponding number of subjects.

Characteristic	**Value/Description**
Number of subjects	8
Number of views	16
Total Number of Frames	100 K
Ethnicity	5 Asian, 3 Caucasian
Gender	6 male, 2 female
Age	20 - 29 (5), 30 - 39 (3)
Facial expressions	neutral, open mouth, smiling
Action types	poking a cheek (open hand) poking a cheek (pointing hand) punching a cheek pushing a cheek with a palm rubbing a cheek pinching a chin touching nose front touching nose from side
Data types	2D hand keypoints 2D face landmarks RGB videos foreground segmentation masks hand-face bounding box 3D mesh for hand and face 3D surface deformations

Note that in this method variant, DefConNet and non-rigid collision handling (6.10) are deactivated. Our dataset contains separate training and testing sequences containing the same kinds of actions. We train our networks on the training sequences of 5 different subjects and conduct the quantitative evaluations on 3 different subjects unseen during the training. For the qualitative comparisons, we show the results of our data recording green studio and indoor sequences captured using a SONY DSC-RX0 camera.

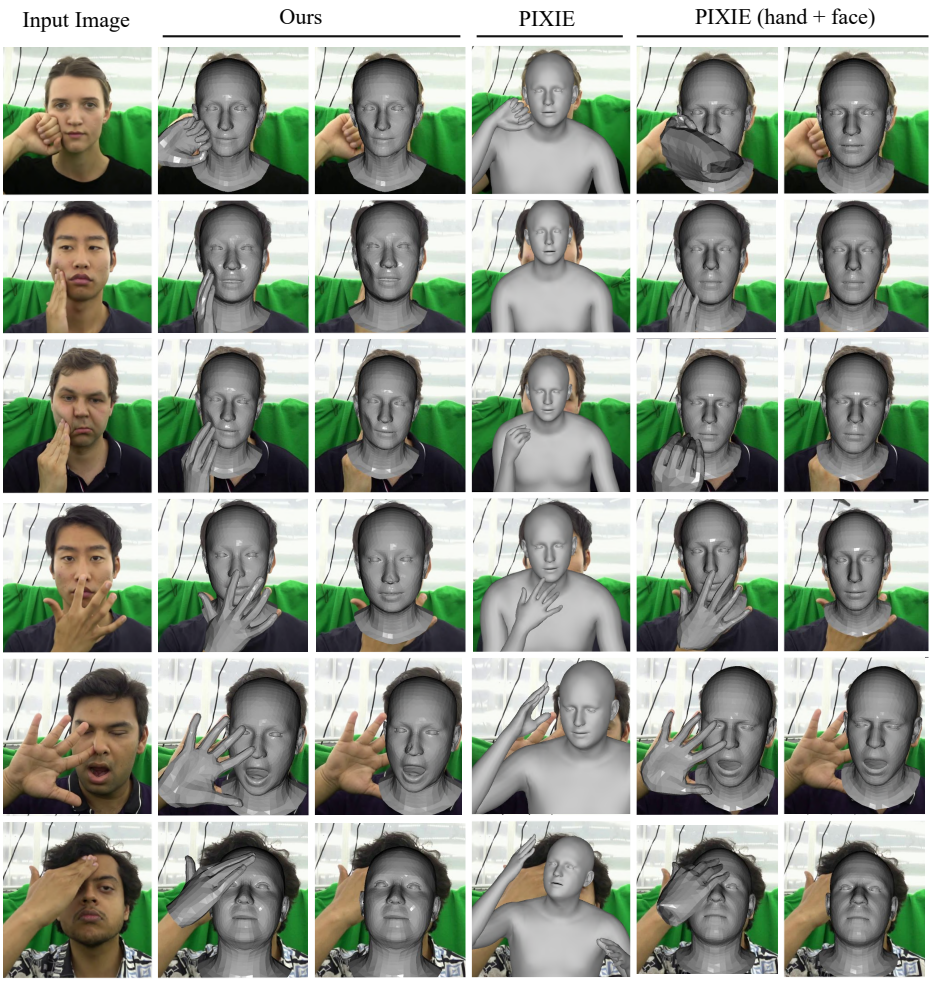


Figure 6.8: Visualisations of the experimental results by our method, PIXIE (Feng et al., 2021a) and hand-face only mode of PIXIE. The PIXIE results (fourth column) frequently lack interactions between the hand and face, resulting in a low touchness ratio (Table 6.2). PIXIE (hand+face) in the fifth column shows collisions and lacks face-hand interactions as the method is agnostic to the latter. Our results (second column) exhibit natural interactions between the hand and face along with plausible face deformations (third column), which are not present in the results of the competing approaches (fourth and sixth columns).

6.5.1 *Implementation and Training Details*

The neural networks were implemented in PyTorch (Paszke et al., 2019). The evaluations and network training were conducted on a computer with an NVIDIA QUADRO RTX 8000 graphics card and AMD EPYC 7502P 32 Core Processor. The training was continued until convergence using

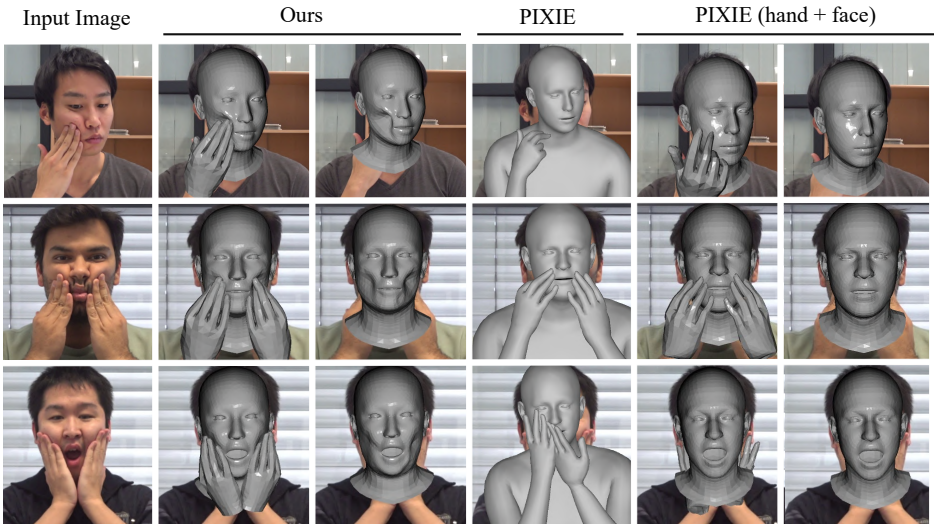


Figure 6.9: Visualisations of the experimental results by our method, PIXIE (Feng et al., 2021a) and hand-face-only mode of PIXIE for indoor scenes. Similar to the case with the green-screen studio (Fig. 6.8), the results in this experimental setup are plausible and represent expressive facial deformations, whereas PIXIE (Feng et al., 2021a) and its slimmed-down version show inaccurate interactions and lack deformations.

Adam optimiser (Kingma and Ba, 2015b) with a learning rate $3 \cdot 10^{-4}$. *DefConNet* models are trained until convergence which takes ≈12 hours. Since our dataset was captured with right-hand and face interactions, we flip the image and the corresponding 3D ground-truth annotations and contact labels horizontally to obtain the input and ground truth for the left hand. For the global fitting optimisation, we set the loss term weights of (6.5), $\lambda_{\boldsymbol{\beta}} = 1 \cdot 10^{-5}$, $\lambda_{\boldsymbol{\Psi}} = 1 \cdot 10^{-3}$, $\lambda_{\dot{\mathbf{V}}} = 3 \cdot 10^{-4}$, $\lambda_{\ddot{\mathbf{V}}} = 3 \cdot 10^{-4}$. For (6.8), we employed the following weights: $\lambda_{\text{touch}} = 0.1$, $\lambda_{\text{col.}} = 1.0$, $\lambda_{\text{depth}} = 3 \cdot 10^{-3}$, $\lambda_{\boldsymbol{\beta}} = 1 \cdot 10^{-5}$, $\lambda_{\dot{\mathbf{V}}} = 3 \cdot 10^{-4}$, $\lambda_{\ddot{\mathbf{V}}} = 3 \cdot 10^{-4}$. As the 2D hand keypoint estimator (Lugaresi et al., 2019) in our method estimates 3D hand key points as well, we utilise them to initialise our hand pose by simply fitting the MANO hand model onto the 3D keypoints using inverse kinematics (Note that this step is optional.).

6.5.2 *Qualitative Evaluations*

Fig. 6.8 and Fig. 6.9 show comparisons of our results with results of PIXIE (Feng et al., 2021a) and its hand+face only version in a studio and an

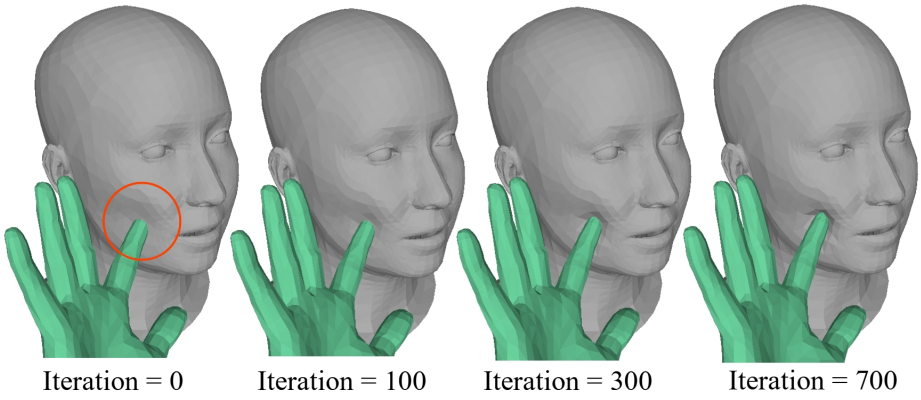


Figure 6.10: Visualisation of the effect of $\mathcal{L}_{\text{col.}}$ (6.10). Starting from the colliding hand and face poses (left-most visualisation), our non-rigid collision loss term effectively resolves the physically implausible inter-penetrations in the course of the optimisation.

indoor scene. Only our method reconstructs face deformations caused by the interactions while showing much more accurate 3D localisations of the hands and face compared to other approaches. In Fig. 6.10, we also show an example visualisation of the non-rigid collision loss (6.10) starting from colliding hand and face positions. While the optimisation progresses, the physically implausible collisions are resolved by plausibly deforming the face surface. Our qualitative results confirm that *Decaf* produces significantly more plausible hand-face interactions and natural face deformations from a single RGB video compared with others.

To assess the generalisability of our *Decaf* across diverse identities and lighting conditions, we evaluate it on in-the-wild images; see Fig. 6.11. The reconstructed 3D shapes show plausible interactions with reasonable facial deformations. Furthermore, the estimated contacts showcased in Fig. 6.12 faithfully mirror the contact regions evident in the input images. As a result, the final reconstructions show plausible hand-to-face interactions guided by the estimated contacts. To further assess the generalisability of our method on unseen actions, we train our networks excluding "poking a cheek (pointing hand)" and "punching a cheek" actions from the training dataset; the results for these actions are illustrated in Fig. 6.13. Our method produces satisfactory results for "poking a cheek (pointing hand)". On the other hand, the exclusion of "punching a cheek" from the training dataset is a highly challenging scenario as no other actions in the training data contain interactions between the back

Figure 6.11: 3D reconstructions on unseen identities in the wild. Our *Decaf* reasonably generalises across different identities and illuminations unseen during the training.

side of the hand and the face. Given that our approach is neural and learning-based, such a substantial deviation from the training set can lead to inaccurate interactions in the results.

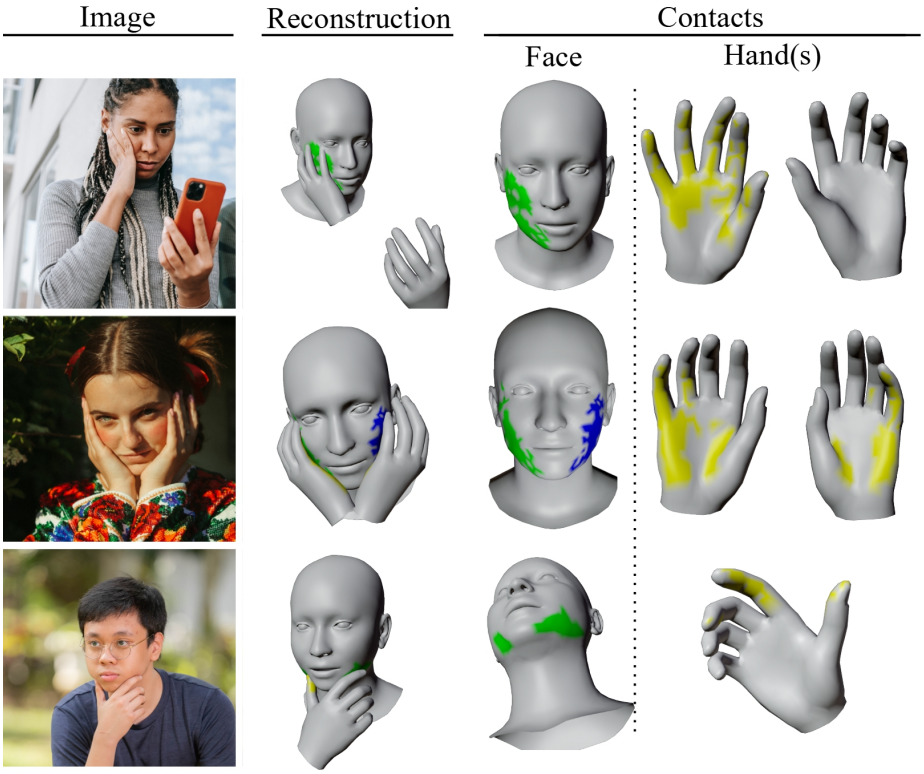


Figure 6.12: Visualisations of the estimated contacts on in-the-wild images. The green and blue colours represent the face contacts regressed by the right- and left-hand DefConNet, respectively (see Fig. 6.2). The yellow colour represents the contact regions on the hand(s). All estimations are reasonable.

6.5.3 *Quantitative Evaluations*

To evaluate our algorithm from various perspectives numerically, we report multiple evaluation metrics. We calculate the 3D per vertex error (PVE) as an indicator of the 3D accuracy as well as the 3D deformation errors for our estimated face deformations. Additionally, we report the metrics of *collision distance*, *non-collision ratio* and *touchness ratio* to quantify the physical plausibility of the reconstructed hands and faces. We also include the *F-Score* to evaluate the overall plausibility of the reconstructions, taking into account both the occurrences of collisions and the correctness of the interactions. The specific details of each metric are elaborated as follows:

- **Per vertex error (PVE)** measures the magnitude of the 3D error by computing the average Euclidean distances between the reconstruction

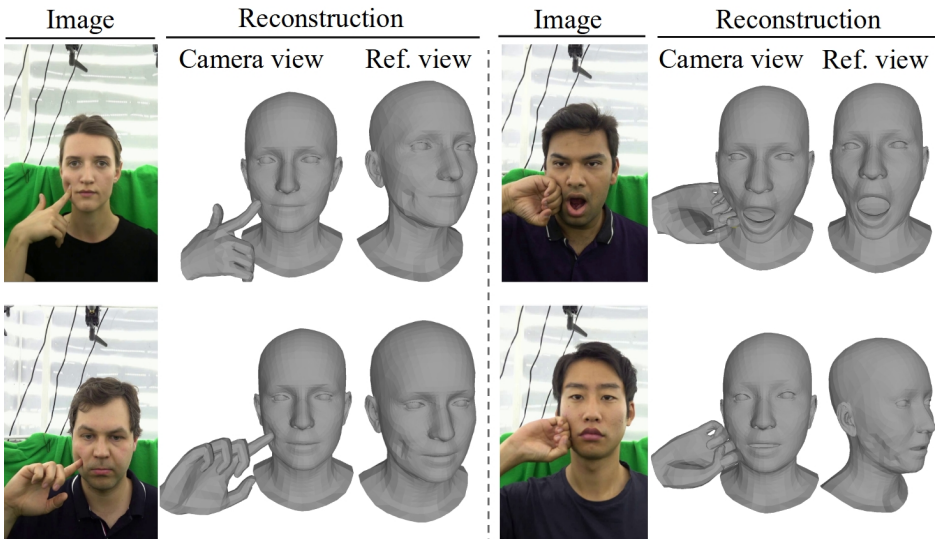


Figure 6.13: 3D reconstructions on actions unseen during the training, *i.e.* (left:) poking a cheek (pointing hand) and (right:) punching a cheek.

and the ground-truth vertices. We report the errors in the camera frame before and after applying a translation on the hand and face that aligns the centroid of the face with the origin of the coordinate frame, denoted as PVE and PVE†, respectively. Hence, PVE† measures the reconstruction quality focusing on the relative position of the hand w.r.t. the head, which is important when judging the accuracy of hands-head interactions.

- **Deformation error (DefE)** measures the magnitude of the error by computing the average Euclidean distances between the estimated per-vertex 3D deformations and their pseudo ground truth. We also report **+DefE** that computes DefE only for deformations with the corresponding ground-truth deformation vectors of norm greater than 5 [mm], *i.e.* when non-negligible interactions are present. Lower DefE and +DefE indicate higher prediction accuracy of the deformations.

- **Collision distance (Col. Dist.)** measures the collision distances averaged over the number of vertices and frames. A lower collision distance indicates a smaller magnitude of collisions throughout the sequence.

- **Non-collision ratio (Non. Col.)** measures the ratio of the frames with no collisions between the hand and face over all sequence frames. A

Table 6.2: Comparisons of the 3D reconstruction accuracy and plausibility of interactions. "†" denotes PVE after applying a translation on both the face and hand that translates the centre of the face mesh to the origin.

	3D Error		Plausibility Measurement			
	PVE↓ [mm]	PVE†↓ [mm]	Col.Dist.↓ [mm]	Non.Col. ↑ [%]	Touchness ↑ [%]	F-Score ↑ [%]
Ours	**11.9**	**9.65**	1.03	83.6	96.6	**89.6**
Ours w/o $\mathcal{L}_{\text{touch}}$	17.4	15.2	6.83	68.7	78.5	73.2
Ours w/o $\mathcal{L}_{\text{col.}}$	15.7	12.9	14.4	59.6	87.7	71.0
Ours w/o $\mathcal{L}_{\text{depth}}$	15.9	13.8	11.0	77.2	85.5	81.1
Benchmark	18.9	17.7	19.3	64.2	73.2	68.4
PIXIE (hand+face)	41.6	26.3	7.04	75.9	75.1	75.5
PIXIE	51.9	39.7	0.11	97.1	51.8	67.6

higher non-collision ratio indicates fewer collisions in the reconstructed sequence.

- **Touchness ratio** measures the ratio of frames over all the frames where contacts between face and hand are present in the prediction when there are face-hand contacts in our ground truth. The hand vertices with the nearest distance from the face surface lower than 5 [mm] are considered in contact. This metric exposes the presence of an artefact, namely the occurrence of face-hand interactions in the input frame while the hand does not make physical contact with the face in the reconstruction. A higher ratio indicates more plausible reconstructions.

- **F-Score** for Non. Col. and touchness ratio are also reported by computing the harmonic mean of the two (as these two metrics are complementary to each other). It is very important to report F-Score, since each of these metrics in isolation is not meaningful (*e.g.* constant presence of hand-face collisions will result in perfect touchness ratio 100%; no presence of interaction throughout the sequence will make the perfect Non. Col. 100%). A higher F-Score indicates a higher plausibility of the interactions in the reconstructions showing fewer occurrences of collisions and incorrect interactions.

3D Error Comparisons. We report PVE in Table 6.2-(left) to evaluate the 3D accuracy of the reconstructed hand and face. Our *Decaf* shows the best performance scoring around 40% less error compared with the

Table 6.3: 3D deformation error comparisons. "+" indicates that DefE was computed only on deformations whose ground-truth deformation vector has a norm greater than 5 [mm]. Note that DefE and +DefE for related methods and benchmarks are computed using zero displacements as only our method outputs the per-vertex deformations (denoted with "*") .

	DefE. [mm]↓	+DefE. [mm]↓
Ours	**0.08**	**2.28**
Ours w/o refinement	0.09	2.35
Benchmark	0.13*	7.28*
PIXIE (hand+face)	0.13*	7.28*
PIXIE	0.13*	7.28*

second best method, benchmark ((Lugaresi et al., 2019) + (Li et al., 2017)). We also report the 3D accuracies of the deformations; DefE and +DefE in Table 6.3. To compute DefE for the related works, we simply provide zero deformations as those methods do not model per-vertex deformations caused by interactions. For both DefE and +DefE, our method shows the lowest errors, *i.e.*about 60% lower errors for DefE and 40% lower errors compared with others.

Plausibility of Interactions. In Table 6.2, we report Col. Dist., Non. Col., Touchness and F-Score. It is very important to show F-Score as Non. Col. and Touchness are *complementary to each other*. Ours show low collision distances while showing quite high *Touchness*, which indicates the highly plausible face-hand interactions that correspond to the input images, thus the best performance in F-Score. In contrast, PIXIE shows extremely low collision distances while showing much worse *Touchness* compared with ours. This is because, in most cases, the reconstructed hand and face are wrongly not interacting with each other when they should be interacting; see Fig. 6.8 for the example reconstructions. The benchmark and PIXIE (hand+face) independently reconstruct the face and hands being agnostic of the interactions of those; therefore, they show quite frequent collisions (high Col. Dist. and low Non. Col.) as well as incorrect interactions (Low Touchness), thus lower F-Score than ours. Given these metrics in Table 6.2 and the qualitative results, *Decaf* shows the most plausible interactions in the reconstructed results compared with the related methods.

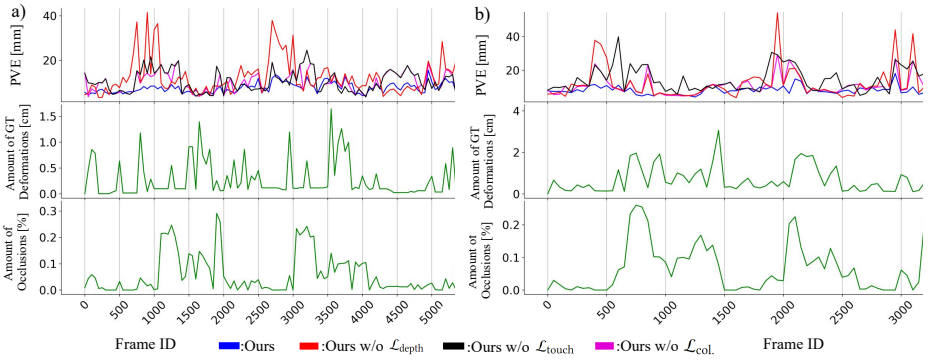


Figure 6.14: PVE plots for two exemplary test sequences (left: woman on top-left in Fig. 6.7; right: man on middle-right in Fig. 6.7) in relation to the degree of occlusions and deformations in the pseudo ground truth. Our full model is affected by the occlusions (the bottom row) substantially less than its ablated versions.

Ablation Studies In Table 6.2, we show the ablation studies of the reconstructions denoted as "Ours w/o $\mathcal{L}_{\text{touch}}$", "Ours w/o $\mathcal{L}_{\text{col.}}$ " and "Ours w/o $\mathcal{L}_{\text{depth}}$ " to assess the importance of each loss term. For both the 3D accuracy and plausibility measurements, removing one loss term results in a severe performance decrease, which confirms all those loss terms contribute to higher 3D localisations and improvement of interaction plausibilities. Additionally, in Table 6.3, we also show the DefE and +DefE without updating the deformations in the final global fitting optimisation stage, *i.e.* direct output from the DefConNet denoted as "Ours w/o refinement". Our final global fitting optimisation improves the estimated deformations from DefConNet, reducing the DefE and +DefE by 10% and 3%.

Fig. 6.14 shows PVE plots for two test sequences from our dataset, highlighting the stability of our results. *Amount of occlusion* stands for the per-frame ratio of face pixels occluded by hand pixels from the camera view and *amount of deformations* signifies the per-frame sums of deformations in the pseudo ground truth. We observe that the ablated versions of our method are starkly influenced by occlusions, which can be recognised with the help of peaks occurring at the frames with the (locally) largest deformations as well as the most significant occlusions. In contrast, our full model is affected by the occlusions substantially less, and its curve has a smaller standard deviation of PVE, which verifies the importance of each loss term.

Table 6.4: Perfomance measurement of our contact estimation component. Our method estimates reasonable contacts on face-hand surfaces only from RGB input, which are integrated into the final global fitting optimisation. The significance of the contacts is validated in Table 6.2.

	F-score ↑	Precision ↑	Recall ↑	Accuracy ↑
face	0.57	0.69	0.49	0.99
hand	0.47	0.62	0.39	0.98

Contact Estimations. To our knowledge, there are no existing works that estimate the dense contacts on hand-face surfaces from RGB inputs. Nonetheless, we report the performance of the contact estimation of our method for comparison on Table 6.4. Note that although estimating contact vertices only from RGB inputs is a highly challenging problem, our *Decaf* estimates reasonable contacts that significantly improve the 3D localisation as validated in Table 6.2.

6.6 DISCUSSIONS AND LIMITATIONS

Our *Decaf* captures plausible 3D deformations along with hand-face interactions solely from a monocular RGB video, effectively reducing unnatural collisions and non-touching artefacts. While our method is the first to address this problem set, it does have certain limitations. Our network learns from a newly created dataset computed using Position-Based Dynamics (PBD) with a skull-skin-distance (SSD) approach combined with the multi-view markerless motion capture setup. PBD is widely utilised in modern physics engines, ensuring that our pseudo-ground truth deformations are plausible. However, it may introduce some discrepancies between the actual deformations and calculated deformations as this PBD-based approach does not integrate visual information such as photometric loss. Nevertheless, we believe this approach to be satisfactorily accurate to obtain plausible deformations although the visual information is not reliable at the interaction regions due to the constant occlusions, which is verified in our qualitative experiments.

Our method employs PCA-based parametric face and hand models. Consequently, the 3D reconstructions of both body parts maintain consis-

tent topology though, as a downside, miss high-frequency details such as wrinkles or blood vessels.

Lastly, our method primarily focuses on handling pushing actions (*e.g.* pushing or poking cheeks). Furthermore, it is important to note that object-hand-face interactions, which fall outside the scope of our research, can be addressed in future studies.

6.7 CONCLUSIONS

Decaf is the first monocular RGB-based approach for deformation-aware 3D hand-face motion capture. Our method captures non-rigid face surface deformations arising from various hand-head interactions. It regards the human head anatomy (*i.e.* skull-skin distance used to calculate non-uniform facial tissue stiffness), detects hand-head contacts and is trained on a new dataset of facial performances. In our experimental evaluation, *Decaf* demonstrates the highest 3D reconstruction (in terms of PVE) and plausibility metrics (in terms of F-score) among all compared methods. Especially significant are the advancement in terms of PVE compared to the most closely related previous method (roughly fourfold error reduction) and qualitative improvements in the estimated 3D geometry, which opens up many possibilities for downstream applications (*e.g.* next-generation telepresence systems).

7

MACS: MASS CONDITIONED 3D HAND AND OBJECT MOTION SYNTHESIS

The preceding chapter introduced the first monocular RGB video-based MoCap method that reconstructs 3D face-hand motions simultaneously with the deformations resulting from them. This approach incorporated explicit modelling of interaction-induced deformations within a learning-based framework. Moreover, the novel VAE-based interaction prior network played a pivotal role in disambiguating the depth of a face and hands during the reconstruction. The chapter also introduced the first 3D deformation dataset with corresponding multi-view videos for hand-face interactions, which was leveraged to train the deformation estimator and the interaction prior.

In this thesis, several physics-based MoCap approaches were introduced. However, it is highly challenging to estimate all physical quantities that are part of the mathematical physics models, given only a monocular video (*e.g.* mass distribution of the subject, friction coefficient of the ground and foot, and more). Consequently, the previous chapters resorted to using average values for these physical parameters, which can introduce certain inaccuracies in the estimates. This indicates the importance of a 3D motion dataset tied to ground truth physical quantities that allows the development of learning-based approaches that regress such physical quantities accurately.

This chapter (published as Shimada et al., 2024) introduces the first approach for synthesising 3D object manipulations with hands conditioned by a physical quantity, namely the mass of the manipulated object. The synthesised motions exhibit behaviour that faithfully adapts to the conditioning mass. This method can be helpful for ML applications as it can generate a 3D motion dataset with a corresponding mass value. Furthermore, the proposed method offers the flexibility of optionally incorporating user-provided object trajectories as inputs while tailoring the motion behaviour based on the conditioning mass. This feature has the potential to reduce the workload of 3D motion designers significantly.

Our experiments confirm that the synthesised motions show a high motion diversity and plausibility in terms of motion dynamics and interactions. Moreover, the proposed method is favoured compared with its benchmark methods in the user study.

7.1 INTRODUCTION

Hand-object interaction plays an important role in our daily lives, involving the use of our hands in a variety of ways such as grasping, lifting, and throwing. It is crucial for graphics applications (*e.g.* AR/VR, avatar communication and character animation) to synthesise or capture physically plausible interactions for their enhanced realism. Therefore, there has been a growing interest in this field of research, and a significant amount of work has been proposed in grasp synthesis (Grady et al., 2021; Karunratanakul et al., 2020; Krug et al., 2010; Li et al., 2007; Taheri et al., 2020), object manipulation (Christen et al., 2022; Ghosh et al., 2023; Mordatch et al., 2012; Ye and Liu, 2012; Zhang et al., 2021c), 3D reconstruction (Corona et al., 2020; Hu et al., 2022; Liu et al., 2021; Mueller et al., 2019; Schroder and Ritter, 2017; Tekin et al., 2019; Wang et al., 2020a), graph refinement (Detry et al., 2010; Pollard and Zordan, 2005; Zhou et al., 2022) and contact prediction (Brahmbhatt et al., 2019).

Because of the high dimensionality of the hand models and inconsistent object shape and topology, synthesising plausible 3D hand-object interaction is challenging. Furthermore, errors of even a few millimetres can cause collisions or floating-object artefacts that immediately convey an unnatural impression to the viewer. Some works tackle the static grasp synthesis task using an explicit hand model (Grady et al., 2021; Krug et al., 2010; Taheri et al., 2020) or an implicit representation (Karunratanakul et al., 2020). However, considering the static frame alone is not sufficient to integrate the method into real-world applications such as AR/VR as it lacks information of the inherent scene dynamics. Recently, several works have been proposed to synthesise the hand and object interactions as a continuous sequence (Christen et al., 2022; Zhang et al., 2021c; Zhou et al., 2022). However, none of the state-of-the-art work explicitly considers an object's mass when generating hand-object interactions. Real-life object manipulation, however, is substantially influenced by the mass of the objects we are interacting with. For example, we tend to grab light

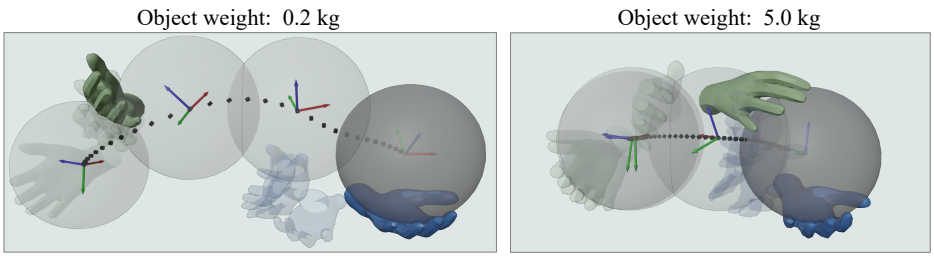


Figure 7.1: Example visualisations of 3D object manipulation synthesised by our method *MACS*. Conditioning object mass values of 0.2kg (left) and 5.0kg (right) are given to the model for the action type "passing from one hand to another". *MACS* plausibly reflects the mass value in the synthesised 3D motions.

objects using our fingertips, whereas with heavy objects oftentimes the entire palm is in contact with the object. Manually creating such animations is tedious work requiring artistic skills. In this chapter, we propose *MACS*, *i.e.* the first learning-based *mass conditioned* object manipulation synthesis method. The generated object manipulation naturally adopts its behaviour depending on the object mass value. *MACS* can synthesise such mass conditioned interactions given a trajectory plus action label (*e.g.* throw or move). The trajectory itself may also be generated conditioned on the action label and mass using the proposed cascaded diffusion model, or alternatively manually specified.

Specifically, given the action label and mass value as conditions, our cascaded diffusion model synthesises the object trajectories as the first step. The synthesised object trajectory and mass value further condition a second diffusion model that synthesises 3D hand motions and hand contact labels. After the final optimisation step, *MACS* returns diverse and physically plausible object manipulation animations. We also demonstrate a simple but effective data capture set-up to produce a 3D object manipulation dataset with corresponding mass values. The contributions of the method are as follows:

- The first approach to synthesise *mass-conditioned* object manipulations in 3D. The setting includes two hands and a single object of varying mass.
- A cascaded denoising diffusion model for generating trajectories of hands and objects allowing different types of conditioning inputs. Our approach can both synthesise new object trajectories and oper-

ate on user-provided trajectories (in this case, the object trajectory synthesis part is skipped).

- A new component for introducing plausible dynamics into user-provided trajectories.

The comprehensive experiments confirm that *MACS* synthesises qualitatively and quantitatively more plausible 3D object manipulations compared with other baselines. *MACS* shows plausible manipulative interactions even for mass values vastly different from those seen during the training.

7.2 RELATED WORK

There has been a significant amount of research in the field of 3D hand-object interaction motion synthesis. Here, we will review some of the most relevant works in this area. Grasp synthesis works are discussed in Sec. 7.2.1 and works that generate hand-object manipulation sequences in Sec. 7.2.2. Lastly, closely related recent diffusion model based synthesis approaches are discussed in Sec. 7.2.3.

7.2.1 *Grasp Synthesis*

Synthesising physically plausible and natural grasps bears a lot of potential downstream applications. Thus, many works in this field have been proposed in computer graphics and vision (Ghosh et al., 2023; Li et al., 2007; Pollard and Zordan, 2005; Ye and Liu, 2012; Zhang et al., 2021c), and robotics community (Krug et al., 2010; Thobbi and Sheng, 2010). ContactOpt (Grady et al., 2021) utilises a differentiable contact model to obtain a plausible grasp from a hand and object mesh. Karunratanakul et al. (2020) proposed a *grasping field* for a grasp synthesis where hand and object surfaces are implicitly represented using a signed distance field. Zhou et al. (2022) proposed a learning-based object grasp refinement method given noisy hand grasping poses. GOAL (Taheri et al., 2022) synthesises a whole human body motion with grasps along with plausible head directions. These works synthesise natural hand grasp on a variety of objects. However, unlike the methods in this class, we

synthesise a sequential object manipulation, changing not only the hand pose but also object positions bearing plausible hand-object interactions.

7.2.2 *Object Manipulation*

Synthesising a sequence for object manipulation is challenging since the synthesised motions have to contain temporal consistency and plausible dynamics considering the continuous interactions. Ghosh et al. (2023) proposed a human-object interaction synthesis algorithm associating the intentions and text inputs. ManipNet (Zhang et al., 2021c) predicts dexterous object manipulations with one/two hands given 6 DoF of hands and object trajectory from a motion tracker. CAMS (Zheng et al., 2023) synthesises hand articulations given a sequence of interacting object positions. Unlike these approaches, our algorithm **synthesises** the 6 DoF of the hands and objects as well as the finger articulations affected by the conditioned mass values. D-Grasp (Christen et al., 2022) is a reinforcement learning-based method that leverages a physics simulation to synthesise a dynamic grasping motion that consists of approaching, grasping and moving a target object. In contrast to D-Grasp, our method consists of a cascaded diffusion model architecture, allowing controllability regarding the object trajectory and having explicit control over the object mass value that influences the synthesised interactions. Furthermore, D-Grasp uses a predetermined target grasp pose and, therefore, does not faithfully adjust its grasp based on the mass value in the simulator unlike ours.

7.2.3 *Diffusion Model based Synthesis*

Recently, diffusion model (Sohl-Dickstein et al., 2015) based synthesis approaches have been receiving growing attention due to their promising results in a variety of research fields, *e.g.* image generation tasks (Ho et al., 2020; Rombach et al., 2022; Saharia et al., 2022), audio synthesis (Kong et al., 2021), motion synthesis (Dabral et al., 2023; Tevet et al., 2023; Yuan et al., 2023; Zhang et al., 2022) and 3D character generation from texts (Poole et al., 2023). MDM (Tevet et al., 2023) shows the 3D human motion synthesis and inpainting tasks from conditional action or text inputs utilising a transformer-based architecture allowing the integration

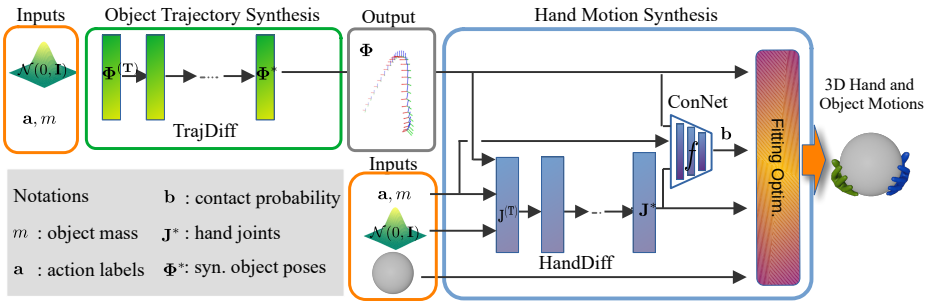


Figure 7.2: **The proposed framework.** The object trajectory synthesis stage accepts as input the conditional mass value m and action label $\mathbf{a}$ along with a Gaussian noise sampled from $\mathcal{N}(0, \mathbf{I})$, and outputs an object trajectory. The hand motion synthesis stage accepts $\mathbf{a}$, m and the synthesised trajectory as conditions along with a Gaussian noise sampled from $\mathcal{N}(0, \mathbf{I})$. ConNet, in this stage, estimates the per-vertex hand contacts from the synthesised hand joints, object trajectory and conditioning values $\mathbf{a}$, m. The final fitting optimisation step returns a set of 3D hand mesh that plausibly interacts with the target object.

of the geometric loss terms during the training. Unlike the existing works in the literature, our method condition the synthesised motions on a physical property, *i.e.* object mass.

7.3 METHOD

Our goal is to synthesise 3D motion sequences of two hands interacting with an object whose mass affects both the trajectory of the object and the way the hands grasp it. The inputs of this method are a conditional scalar mass value and optionally a one-hot coded action label and/or a manually drawn object trajectory. Our method synthesises a *motion* represented as N successive pairs of 3D hands and object poses. To this end, we employ denoising diffusion models (DDM) (Sohl-Dickstein et al., 2015) for 3D hand motion and object trajectory synthesis; see Fig. 7.2 for the overview. We first describe our mathematical modelling and assumptions in Sec. 7.3.1. In Secs. 7.3.2 and 7.3.3, we provide details of our hand motion synthesis network *HandDiff* and trajectory synthesis algorithm *TrajDiff*, respectively. We describe the method to synthesise the 3D motions given user input trajectory in Sec. 7.3.3.2.

7.3.1 *Assumptions, Modelling and Preliminaries*

We assume that the target object is represented as a mesh. 3D hands are represented with a consistent topology, which is described in the following paragraph.

HAND AND OBJECT MODELLING To represent 3D hands, we employ the hand model from GHUM (Xu et al., 2020) which is a nonlinear parametric model learned from large-scale 3D human scans. The hand model from GHUM defines the 3D hand mesh as a differentiable function $\mathcal{M}(\boldsymbol{\tau}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\beta})$ of global root translation $\boldsymbol{\tau} \in \mathbb{R}^3$, global root orientation $\boldsymbol{\phi} \in \mathbb{R}^6$ represented in 6D rotation representation (Zhou et al., 2019), pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{90}$ and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{16}$. We employ two GHUM hand models to represent left and right hands, which return hand vertices $\mathbf{v} \in \mathbb{R}^{3l}$ ($l = 1882 = 941 \cdot 2$) and 3D hand joints $\mathbf{j} \in \mathbb{R}^{3K}$ ($K = 42 = 21 \cdot 2$). The object pose is represented by its 3D translation $\boldsymbol{\tau}_{\text{obj.}} \in \mathbb{R}^3$ and rotation $\boldsymbol{\phi}_{\text{obj.}} \in \mathbb{R}^6$. Our method *MACS* synthesises N successive (i) 3D hand motions represented by the hand vertices $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\} \in \mathbb{R}^{N \times 3l}$ and hand joints $\mathbf{J} = \{\mathbf{j}_1, ..., \mathbf{j}_N\} \in \mathbb{R}^{N \times 3K}$, and (ii) optionally object poses

$$\boldsymbol{\Phi} = \{\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_N\} \in \mathbb{R}^{N \times (3+6)}, \tag{7.1}$$

where $\boldsymbol{\Phi}_i = [\boldsymbol{\tau}_{\text{obj.},i}, \boldsymbol{\phi}_{\text{obj.},i}]$. The object pose is defined in a fixed world frame $\mathcal{F}$, and the global hand translations are represented relative to the object centre position. The global hand rotations are represented relative to $\mathcal{F}$.

DENOISING DIFFUSION MODEL The recently proposed Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) has shown compelling results both in image synthesis tasks and in motion generation tasks (Tevet et al., 2023). Compared to other existing generative models (*e.g.* VAE (Sohn et al., 2015) or GAN (Goodfellow et al., 2014)) that are often employed for motion synthesis tasks, the training of DDPM is simple, as it is not subject to the notorious mode collapse while generating motions of high quality and diversity.

Following the formulation by Ho et al. (2020), the forward diffusion process is defined as a Markov process adding Gaussian noise in each

step. The noise injection is repeated T times. Next, let $\mathbf{X}^{(0)}$ be the original ground-truth (GT) data (without noise). Then, the forward diffusion process is defined by a distribution $q(\cdot)$:

$$q\left(\mathbf{X}^{(1:T)} \mid \mathbf{X}^{(0)}\right) = \prod_{t=1}^{T} q\left(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t-1)}\right), \tag{7.2}$$

$$q\left(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{X}^{(t)} \mid \sqrt{1-\beta_t}\mathbf{X}^{(t-1)}, \beta_t \mathbf{I}\right), \tag{7.3}$$

where β_t are constant hyperparameters (scalars) that are fixed per each diffusion time step t. Using a reparametrisation technique, we can sample $\mathbf{X}^{(t)}$ using the original data $\mathbf{X}^{(0)}$ and standard Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$:

$$\mathbf{X}^{(t)} = \sqrt{\alpha_t}\mathbf{X}^{(0)} + \sqrt{1-\alpha_t}\epsilon, \tag{7.4}$$

where $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$. The network is trained to reverse this process by denoising on each diffusion time step starting from a standard normal distribution $\mathbf{X}^{(T)} \sim \mathcal{N}(0, I)$:

$$p\left(\mathbf{X}^{(0:T)}\right) = p\left(\mathbf{X}^{(T)}\right) \prod_{t=1}^{T} p\left(\mathbf{X}^{(t-1)} \mid \mathbf{X}^{(t)}\right), \tag{7.5}$$

where $p\left(\mathbf{X}^{(t-1)} \mid \mathbf{X}^{(t)}\right)$ denotes the conditional probability distribution estimated from the network output. From Eq. (7.5), we obtain the meaningful generated result $\mathbf{X}^{*}$ after T times of denoising process. that follows the data distribution of the training dataset.

In the formulation of DDPM (Ho et al., 2020), the network is trained to predict the added noises on the data for the reverse diffusion process. The *simple* loss term is formulated as

$$\mathcal{L}_{\text{simple}} = E_{\epsilon, t\sim[1,T]}\left[\left\|\epsilon - \epsilon_\theta\left(\mathbf{X}^{(t)}, t, c\right)\right\|_2^2\right], \tag{7.6}$$

where c denotes an optional conditioning vector. The loss term of Eq. (7.6) drives the network ϵ_θ towards predicting the added noise. Training the network with Eq. (7.6) alone already generates highly diverse motions.

In our case, $\mathbf{X}^{*}$ represents sequences of 3D points corresponding to the synthesised motion trajectories (for hands and objects). Unfortunately, Eq. (7.6) alone often leads to artefacts in the generated sequences such as joint jitters and varying bone lengths when applied to motion synthesis.

To improve the plausibility of the generated results, Dabral et al. (2023) proposed an algorithm to integrate the explicit geometric loss terms into the training of DDPM. At an arbitrary diffusion time step t, we can obtain the approximated original data $\hat{\mathbf{X}}^{(0)}$ using the estimated noise from ϵ_θ instead of ϵ in Eq. (7.4) and solving for $\hat{\mathbf{X}}^{(0)}$:

$$\hat{\mathbf{X}}^{(0)} = \frac{1}{\sqrt{\alpha}}\mathbf{X}^{(t)} - \left(\sqrt{\frac{1}{\alpha} - 1}\right) \epsilon_\theta \left(\mathbf{X}^{(t)}, t, c\right). \tag{7.7}$$

During the training, geometric penalties can be applied on $\hat{\mathbf{X}}^{(0)}$ so as to prevent the aforementioned artefacts. In the following sections, we follow the mathematical notations of DDPM literature (Dabral et al., 2023; Ho et al., 2020) as much as possible. The approximated set of hand joints and object poses obtained from Eq. (7.7) are denoted $\hat{\mathbf{J}}^{(0)}$ and $\hat{\mathbf{\Phi}}^{(0)}$, respectively. Similarly, the synthesised set of meaningful hand joints and object poses obtained from the reverse diffusion process Eq. (7.5) are denoted $\mathbf{J}^*$ and $\mathbf{\Phi}^*$, respectively.

7.3.2 *Hand 3D Motion Synthesis*

Our DDPM-based architectures *HandDiff* $\mathcal{H}(\cdot)$ and *TrajDiff* $\mathcal{T}(\cdot)$ are based on the stable diffusion architecture (Rombach et al., 2022) with simple 1D and 2D convolution layers. During the training, we follow the formulation of Dabral et al. (2023) described in Sec. 7.3.1 to introduce geometric penalties on $\hat{\mathbf{J}}^{(0)} \in \mathbb{R}^{N\times 3K}$ and $\hat{\mathbf{\Phi}}^{(0)} \in \mathbb{R}^{N\times 9}$ combined with the simple loss described in Eq. (7.6).

HAND KEYPOINTS SYNTHESIS In this stage, we synthesise a set of 3D hand joints and per-vertex hand contact probabilities. Knowing the contact positions on hands substantially helps to reduce the implausible "floating object" artefacts of the object manipulation (see Sec.7.5 for the ablations). The synthesised 3D hand joints and contact information are further sent to the final fitting optimisation stage where we obtain the final hand meshes considering the plausible interactions between the hands and the object.

Our diffusion model based *HandDiff* $\mathcal{H}(\cdot)$ accepts as inputs a 3D trajectory $\mathbf{\Phi} \in \mathbb{R}^{N\times(3+6)}$ and mass scalar value m where N is the number of

frames of the sequence. From the reverse diffusion process of $\mathcal{H}(\cdot)$, we obtain the synthesised set of 3D joints $\mathbf{J}^* \in \mathbb{R}^{N\times 3K}$. $\mathbf{\Phi}$ can be either synthesised by *TrajDiff* $\mathcal{T}(\cdot)$ (Sec. 7.3.3.1) or manually provided (Sec. 7.3.3.2). Along with the set of 3D hand joint positions, the 1D convolution-based *ConNet* $f(\cdot)$ also estimates the contact probabilities $\mathbf{b} \in \mathbb{R}^{N\times l}$ on the hand vertices from the hand joint and object pose sequence with a conditioning vector $\mathbf{c}$ that consists of a mass value m and an action label $\mathbf{a}$. *ConNet* $f(\cdot)$ is trained using a binary cross entropy (BCE) loss with the GT hand contact labels $l_{\text{con.}}$:

$$\mathcal{L}_{\text{con.}} = \text{BCE}(f(\mathbf{J}^{(0)}, \mathbf{\Phi}^{(0)}, \mathbf{c}), l_{\text{con.}}), \tag{7.8}$$

where $\mathbf{J}^{(0)}$ and $\mathbf{\Phi}^{(0)}$ denotes a set of GT 3D hand joints and GT object poses, respectively. At test time, *ConNet* estimates the contact probabilities from the synthesised 3D hand joints and object positions conditioned on $\mathbf{c}$. The estimated contact probabilities $\mathbf{b}$ are used in the subsequent *fitting optimisation* step, to increase the plausibility of the hand and object interactions.

The objective $\mathcal{L}_{\text{H}}$ for the training of *HandDiff* reads:

$$\mathcal{L}_{\text{H}} = \mathcal{L}_{\text{simple}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}}, \tag{7.9}$$

where $\mathcal{L}_{\text{simple}}$ is computed following Eq. (7.6) and

$$\mathcal{L}_{\text{geo}} = \lambda_{\text{rec.}}\mathcal{L}_{\text{rec.}} + \lambda_{\text{vel.}}\mathcal{L}_{\text{vel.}} + \lambda_{\text{acc}}\mathcal{L}_{\text{acc.}} + \lambda_{\text{blen}}\mathcal{L}_{\text{blen.}}. \tag{7.10}$$

$\mathcal{L}_{\text{rec.}}$, $\mathcal{L}_{\text{vel.}}$ and $\mathcal{L}_{\text{acc.}}$ are loss terms to penalise the positions, velocities and accelerations of the synthesised hand joints, respectively:

$$\mathcal{L}_{\text{rec.}} = \|\hat{\mathbf{J}}^{(0)} - \mathbf{J}^{(0)}\|_2^2, \tag{7.11}$$

$$\mathcal{L}_{\text{vel.}} = \|\hat{\mathbf{J}}^{(0)}_{\text{vel.}} - \mathbf{J}^{(0)}_{\text{vel.}}\|_2^2, \tag{7.12}$$

$$\mathcal{L}_{\text{acc.}} = \|\hat{\mathbf{J}}^{(0)}_{\text{acc.}} - \mathbf{J}^{(0)}_{\text{acc.}}\|_2^2, \tag{7.13}$$

where $\hat{\mathbf{J}}^{(0)}$ is an approximated set of hand joints from Eq. (7.7) and $\mathbf{J}^{(0)}$ denotes a set of GT hand joints. $\hat{\mathbf{J}}^{(0)}$ and $\mathbf{J}^0$ with the subscripts "vel."

and "acc." represent the velocities and accelerations computed from their positions, respectively.

$\mathcal{L}_{\text{blen.}}$ penalises incorrect bone lengths of the hand joints using the function $d_{\text{blen}} : \mathbb{R}^{N\times 3K} \rightarrow \mathbb{R}^{N\times K}$ that computes bone lengths of hands given a sequence 3D hand joints of N frames:

$$\mathcal{L}_{\text{blen.}} = \|d_{\text{blen}}(\hat{\mathbf{J}}^{(0)}) - d_{\text{blen}}(\mathbf{J}^{(0)})\|_2^2. \tag{7.14}$$

At test time, we obtain a set of 3D hand joints $\mathbf{J}^*$ using the denoising process detailed in Eq. (7.5) given a Gaussian noise $\sim N(0, \mathbf{I})$, and a set of per-vertex contact labels.

FITTING OPTIMISATION Once the 3D hand joint sequence $\mathbf{J}^*$ is synthesised from the trained $\mathcal{H}$, we solve an optimisation problem to fit GHUM hand models to $\mathbf{J}^*$. We use a threshold of $\mathbf{b} > 0.5$ to select the effective contacts from the per-vertex contact probability obtained in the previous step. Let $\mathbf{b}^n_{\text{idx}} \subset [\![1, L]\!]$ be the subset of hand vertex indices with effective contacts on the n-th frame. The objectives are written as follows:

$$\underset{\boldsymbol{\tau},\boldsymbol{\phi},\boldsymbol{\theta}}{argmin}(\lambda_{\text{data}}\mathcal{L}_{\text{data}} + \lambda_{\text{touch}}\mathcal{L}_{\text{touch}} + \lambda_{\text{col.}}\mathcal{L}_{\text{col.}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}}). \tag{7.15}$$

$\mathcal{L}_{\text{data}}$ is a data term to minimise the Euclidean distances between the GHUM ($\mathbf{J}$) and the synthesised hand joint key points ($\mathbf{J}^*$):

$$\mathcal{L}_{\text{data}} = \|\mathbf{J} - \mathbf{J}^*\|_2^2. \tag{7.16}$$

$\mathcal{L}_{\text{touch}}$ is composed of two terms. The first term reduces the distances between the contact hand vertices and their nearest vertices $\mathbf{P}$ on the object to improve the plausibility of the interactions. The second term takes into account the normals of the object and hands, which also enhances the naturalness of the grasp by minimising the cosine similarity $s(\cdot)$ between the normals of the contact hand vertices $\mathbf{n}$ and the normals of their nearest vertices of the object $\hat{\mathbf{n}}$.

$$\mathcal{L}_{\text{touch}} = \sum_{i=1}^{N} \sum_{j\in\mathbf{b}^i_{\text{idx}}} \left\|\mathbf{V}^j_i - \mathbf{P}^j_i\right\|_2^2 + \sum_{i=1}^{N} \sum_{i\in\mathbf{b}_{\text{idx}}} (1 - s(\mathbf{n}^j_i, \hat{\mathbf{n}}^j_i)), \tag{7.17}$$

where the subscript i denotes i-th sequence frame and the superscript j denotes the index of the vertex with the effective contact. $\mathcal{L}_{\text{col.}}$ reduces the collisions between the hand and object by minimising the penetration distances. Let $\mathcal{P}^n \subset [\![1, U]\!]$ be the subset of hand vertex indices with collisions on n-th frame. Then we define

$$\mathcal{L}_{\text{col.}} = \sum_{i=1}^{N} \sum_{j \in \mathcal{P}^n} \left\| \mathbf{V}_i^j - \mathbf{P}_i^j \right\|_2^2. \tag{7.18}$$

$\mathcal{L}_{\text{prior}}$ is a hand pose prior term that encourages the plausibility of the GHUM hand pose by minimising the pose vector $\boldsymbol{\theta}$ of the GHUM parametric model:

$$\mathcal{L}_{\text{prior}} = \|\boldsymbol{\theta}\|_2^2. \tag{7.19}$$

With all these loss terms combined, our final output shows a highly plausible hand and object interaction sequence. The effectiveness of the loss terms is shown in our ablative study (Sec. 7.5.2).

7.3.3 *Object Trajectory Generation*

The input object trajectory for *HandDiff* can be provided in two ways, (1) synthesising 3D trajectory by *TrajDiff* (Sec.7.3.3.1) or (2) providing a manual trajectory (Sec. 7.3.3.2). The former allows generating an arbitrary number of hands-object interaction motions conditioned on mass values and action labels, which can contribute to a large-scale dataset generation for machine learning applications. The latter allows for tighter control of the synthesised motions, which are still conditioned on an object's mass value but restricted to the provided trajectory.

7.3.3.1 *Object Trajectory Synthesis*

To provide a 3D object trajectory to *HandDiff*, we introduce a diffusion model-based architecture *TrajDiff* that synthesises an object trajectory given a mass value m and an action label $\mathbf{a} \in \mathbb{R}^6$ encoded as a one-hot vector. We observed that directly synthesising a set of object rotation values causes jitter artefacts. We hypothesise that this

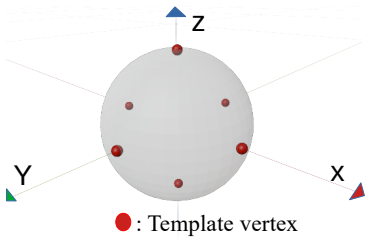


Figure 7.3: Definition of the template vertices.

issue comes from simultaneously synthesising two aspects of a pose, translation and rotation, each having a different representation. As a remedy, we propose to represent both the translation and rotation as 3D coordinates in a Cartesian coordinate system. Specifically, we first synthesise the *reference vertex positions* $\mathbf{P}_{\text{ref}}$ on the object surface defined in the object reference frame, and register them to the predefined *template vertex positions* $\mathbf{P}_{\text{temp}}$ to obtain the rotation of the object. We define 6 template vertices as shown in Fig. 7.3. *TrajDiff* thus synthesises a set of reference vertex positions $\mathbf{P}_{\text{ref}} \in \mathbb{R}^{N \times q}$ where $q = 18 (= 6 \times 3)$ that are defined in the object centre frame along with a set of global translations. We then apply Procrustes alignment between $\mathbf{P}_{\text{ref.}}$ and $\mathbf{P}_{\text{temp.}}$ to obtain the object rotations. The objective of *TrajDiff* is defined as follows:

$$\mathcal{L}_{\mathcal{T}} = \mathcal{L}_{\text{simple}} + \lambda_{\text{geo.}}(\lambda_{\text{rec.}}\mathcal{L}_{\text{rec.}} + \lambda_{\text{vel.}}\mathcal{L}_{\text{vel.}} + \lambda_{\text{acc.}}\mathcal{L}_{\text{acc.}} + \lambda_{\text{ref.}}\mathcal{L}_{\text{ref.}}). \quad (7.20)$$

$\mathcal{L}_{\text{rec.}}$, $\mathcal{L}_{\text{vel.}}$ and $\mathcal{L}_{\text{acc.}}$ follow the definitions given in Eqs. (7.11), (7.12) and (7.13), where $\mathbf{J}^{(0)}$ is replaced with GT 3D object poses whose rotation is represented by the reference vertex positions instead of 6D rotation. $\mathcal{L}_{\text{ref}}$ is defined as:

$$\mathcal{L}_{\text{ref}} = \|\hat{\mathbf{P}}^{(0)}_{\text{ref}} - \mathbf{P}^{(0)}_{\text{ref}}\|_2^2 + \|d_{\text{rel}}(\hat{\mathbf{P}}^{(0)}_{\text{ref}}) - d_{\text{rel}}(\mathbf{P}^{(0)}_{\text{ref}})\|_2^2. \quad (7.21)$$

The first term of $\mathcal{L}_{\text{ref}}$ penalises the Euclidean distances between the approximated reference vertex positions $\hat{\mathbf{P}}^{(0)}_{\text{ref}}$ of Eq. (7.7) and the GT reference vertex positions $\mathbf{P}^{(0)}_{\text{ref}}$. The second term of $\mathcal{L}_{\text{ref}}$ penalises the incorrect Euclidean distances of the approximated reference vertex positions relative to each other. To this end we use a function $d_{\text{rel}} : \mathbb{R}^{N \times 3q} \rightarrow \mathbb{R}^{N \times q'}$, where $q' = \binom{q}{2}$, which computes the distances between all the input vertices pairs on each frame.

The generated object trajectory responds to the specified masses. Thus, the motion range and the velocity of the object tend to be larger for smaller masses. In contrast, with a heavier object, the trajectory shows slower motion and a more regulated motion range.

7.3.3.2 *User-Provided Object Trajectory*

Giving the user control over the output in synthesis tasks is crucial for downstream applications such as character animations or avatar generation. Thanks to the design of our architecture that synthesises 3D hand motions and hand contacts from a mass value and an object trajectory, a manually drawn object trajectory can also be provided to our framework as an input. However, manually drawing an input 3D trajectory is not straightforward, as it must consider the object dynamics influenced by the mass. For instance, heavy objects will accelerate and/or decelerate much slower than lighter ones. Drawing such trajectories is tedious and often requires professional manual labour. To tackle this issue, we introduce a module that accepts a (user-specified) trajectory with an arbitrary number of points along with the object's mass, and outputs a *normalised target trajectory (NTT)*. NTT is calculated from the input trajectory based on the intermediate representation that we call *vector of ratios*, see Fig. 7.4 for the overview. First, the input (user-specified) trajectory is re-sampled uniformly to $N_{fix} = 720$ points and passed to *RatioNet*, which for each time step estimates the distance travelled along the trajectory normalised to the range $[0, 1]$ (*e.g.* the value of 0.3 means that the object travelled 30% of the full trajectory within the given time step). The NTT from this stage is further sent to the *Hand Motion Synthesis* stage to obtain the final hand and object interaction motions. We next explain 1) the initial uniform trajectory re-sampling and 2) the intermediate ratio update approach.

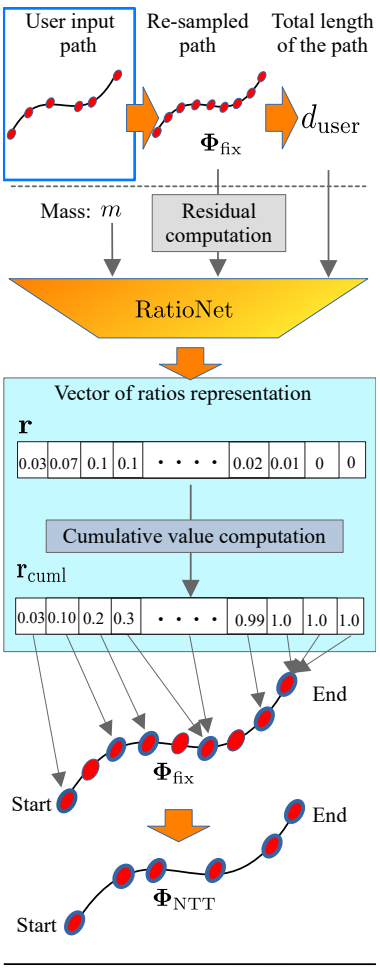


Figure 7.4: Schematic visualisation of the user input trajectory processing stage.

Uniform Input Trajectory Re-sampling To abstract away the variability of the number of points in the user-provided trajectory of N_{user} points, we first interpolate it into a path Φ_{fix} of length N_{fix} points. Note that N_{user} is not fixed and can vary. We also compute the total path length d_{user} that is used as one of the inputs to the RatioNet network (elaborated in the next paragraph):

$$d_{\text{user}} = \sum_{i=1}^{N_{\text{fix}}-1} \|\mathbf{\Phi}_{\text{fix}}^{i} - \mathbf{\Phi}_{\text{fix}}^{i+1}\|^2, \tag{7.22}$$

where $\mathbf{\Phi}_{\text{fix}}^{i}$ denotes the i-th object position in $\mathbf{\Phi}_{\text{fix}}$.

Intermediate Ratio Updates Estimation. From the normalised object path Φ_{fix}, a total distance of the path d_{user}, and mass m, we obtain the information of the object location in each time step using a learning-based approach. To this end, we introduce a MLP-based network *RatioNet* $R(\cdot)$ that estimates the location of the object along the path Φ_{fix} encoded as a ratio starting from the beginning, Specifically, *RatioNet* accepts the residual of Φ_{fix} denoted as $\bar{\Phi}_{\text{fix}}$, a mass scaler value and d_{user} and outputs a vector $\mathbf{r} \in \mathbb{R}^N$ that contains the update of the ratios on the path for each time step:

$$\mathbf{r} = R(\bar{\Phi}_{\text{fix}}, m, d_{\text{user}}). \tag{7.23}$$

Next, we obtain the cumulative ratios $\mathbf{r}_{cuml}$ from $\mathbf{r}$ starting from the time step 0 to the end of the frame sequence. Finally, the NTT $\mathbf{\Phi}_{\text{NTT}} = [\mathbf{\Phi}_{\text{NTT}}^{0}, ..., \mathbf{\Phi}_{\text{NTT}}^{N}]$ at time step t is obtained as:

$$\mathbf{\Phi}_{\text{NTT}}^{t} = \Phi_{\text{fix}}^{id}, \text{ with } id = \text{round}(r_{cum}^{t} \cdot N_{\text{fix}}), \tag{7.24}$$

where id and "$\cdot$" denote the index of Φ_{fix}, and multiplication, respectively. *RatioNet* is trained with the following loss function $\mathcal{L}_{\text{ratio}}$:

$$\mathcal{L}_{\text{ratio}} = \|\mathbf{r} - \hat{\mathbf{r}}\|_2^2 + \|\mathbf{r}_{\text{vel}} - \hat{\mathbf{r}}_{\text{vel}}\|_2^2 + \|\mathbf{r}_{\text{ac.}} - \hat{\mathbf{r}}_{\text{acc}}\|_2^2 + \mathcal{L}_{one}, \tag{7.25}$$

$$\mathcal{L}_{one} = \|(\sum_{i=1}^{N} \mathbf{r}^{i}) - 1\|_2^2, \tag{7.26}$$

where $\hat{\mathbf{r}}$ denotes the GT ratio updates. Note that all terms in Eq. (7.25) have the same weights. The subscripts "vel." and "acc." represent the velocity and accelerations of $\mathbf{r}$ and $\hat{\mathbf{r}}$, respectively. $\mathcal{L}_{one}$ encourages *RatioNet* to estimate the sum of the ratio updates to be 1.0.

7.3.4 *Network Architecture*

We employ the Unet-based diffusion model networks from Ho et al. (2020) for our *TrajDiff* and *HandDiff*. *HandDiff* uses four sets of 2D convolutional residual blocks for the encoder and decoder architecture. *TrajDiff* is composed of two sets of residual blocks of 1D convolution layers instead of 2D convolutions. The number of kernels at its output 1D convolutional layer is set to 21 which corresponds to the dimensionality of the object pose. *ConNet* consists of three-1D convolutional layers with ELU and a sigmoid activation for its hidden layers and output layer, respectively. Similarly, *RatioNet* is composed of three-layer MLP with ELU and sigmoid activation functions in the hidden and output layers, respectively. Starting from the input layer, the output layer dimensions are 1024, 512 and 180.

7.4 DATASET

Figure 7.5: Image of our markered sphere and recording example.

Since there exists no 3D hand and object interaction motion dataset with corresponding object mass values of the objects, we reconstruct such motions using 8 synchronised Z-CAM E2 cameras of 4K resolution and 50 fps. As target objects, we use five plastic spheres of the same radius 0.1[m]. We fill them with different materials of different densities to prepare the objects of the same volume and different weights, *i.e.* $0.175, 2.0, 3.6, 3.9, 4.9$ kg. Each sphere is filled entirely so that its centre of mass does not shift as the object is moved around. Five different subjects are asked to perform five different actions manipulating the object: (1) vertical throw and catch, (2) passing from one hand to another, (3) lifting up and down, (4) moving the object horizontally, and (5) drawing a circle. The subjects perform each action using both their hands while standing in front of the cameras and wearing coloured wristbands (green for the right wrist and yellow for the left wrist), which are later used to classify handedness. The recordings from the multi-view setup were further used to reconstruct the 3D hand and object motions, totalling

Table 7.1: Diversity and multimodality for the hand and trajectory synthesis compared to the ground truth.

	Hand synthesis		Trajectory synthesis	
	Diversity ↑	Multimodality ↑	Diversity ↑	Multimodality ↑
GT	$9.984^{\pm 0.36}$	$7.255^{\pm 0.32}$	$10.041^{\pm 0.28}$	$7.374^{\pm 0.29}$
Ours	$\mathbf{9.606}^{\pm \mathbf{0.33}}$	$\mathbf{7.07}^{\pm \mathbf{0.30}}$	$\mathbf{11.01}^{\pm \mathbf{0.37}}$	$\mathbf{8.05}^{\pm \mathbf{0.33}}$
VAE	$8.350^{\pm 0.42}$	$6.0465^{\pm 0.34}$	$9.584^{\pm 0.47}$	$7.696^{\pm 0.43}$
VAEGAN	$7.821^{\pm 0.27}$	$5.887^{\pm 0.26}$	$8.428^{\pm 0.29}$	$6.285^{\pm 0.30}$

110k frames. The details of the capture and reconstruction processes are described in the following text.

HAND MOTION RECONSTRUCTION To reconstruct 3D hand motions, we first obtain 2D hand key points from all the camera views using MediaPipe (Lugaresi et al., 2019). We then fit GHUM hand models (Xu et al., 2020) for both hands on each frame by solving 2D keypoint reprojection-based optimisation with the known camera intrinsics and extrinsic combining with a collision loss term (Eq. (7.18)), a pose prior loss (Eq. (7.19)) and a shape regulariser term that minimises the norm of the shape parameter β of the GHUM hand parametric model.

OBJECT TRAJECTORY RECONSTRUCTION We place around 50 ArUco markers of the size 1.67×1.67 cm on each sphere for the tracking optimisation (see Fig. 7.5 for the example of our tracking object). The marker positions in the image space are tracked using the OpenCV (Bradski, 2000) library. The 3D object positions on each frame are obtained by solving the multi-view 2D reprojection-based optimisation.

7.5 EXPERIMENTS

To the best of our knowledge, there exists no other work that addresses the hand object manipulation synthesis conditioned on mass. Therefore, we compare our method mainly with two baseline methods which, similarly to our method, employ an encoder-decoder architecture, but which

Table 7.2: Physical plausibility measurement of our full model and its trimmed versions *vs* VAE and VAE-GAN.

	m_{col} [%] ↑	m_{dist} [mm] ↓	m_{touch} [%] ↓
Ours	**97.84**	**0.041**	**1.97**
Ours w/o $\mathcal{L}_{\text{touch}}$	100.0	0.0	63.3
Ours w/o $\mathcal{L}_{\text{col.}}$	38.41	0.296	1.88
VAE	97.2	0.055	2.80
VAE-GAN	96.03	0.058	2.03

Table 7.3: Wasserstein distances between the acceleration distributions ("acc. dist") of the generated motions and ground-truth motions. Combining both $\mathcal{L}_{\text{vel}}$ and $\mathcal{L}_{\text{acc}}$ shows the highest plausibility in terms of the accelerations.

	Ours	Ours w/o $\mathcal{L}_{\text{vel}}$	Ours w/o $\mathcal{L}_{\text{acc}}$
acc. dist. ↓	**7.35**	26.4	11.2

are based on the popular methods VAE (Kingma and Welling, 2014) and VAEGAN (Yu et al., 2019). Specifically, the VAE baseline uses the same diffusion model architecture as our method, but we add a reparameterisation layer (Kingma and Welling, 2014) and remove the skip connections between the encoder and the decoder. The VAEGAN baseline shares the same architecture of the generator, while the discriminator network consists of three 1D convolution layers and two fully connected layers at the output of the network, and we use ELU activation in the discriminator (Clevert et al., 2015). The generator and discriminator networks are conditioned by the same conditioning vector. In all the following experiments, we will refer to our proposed method as *Ours* and to the baselines as *VAE* and *VAEGAN*.

7.5.1 *Training and Implementation Details*

All the networks are implemented in TensorFlow (Abadi et al., 2015) and trained with 1 GPU Nvidia Tesla V100 until convergence. The training of *HandDiff*, *TrajDiff*, *ConNet* and *RatioNet* takes 5 hours, 3 hours, 2 hours and 2 hours, respectively. We set the loss term weights of Eq. (7.10) and (7.20) to $\lambda_{\text{rec.}} = 1.0$, $\lambda_{\text{vel.}} = 5.0$ and $\lambda_{\text{acc.}} = 5.0$. $\lambda_{\text{blen.}}$ of Eq. (7.10) and λ_{ref} of

Table 7.4: Wasserstein distances between the acceleration distributions ("acc. dist") of the generated and ground-truth motions.

	0.175 [kg]	2.0 [kg]	3.6 [kg]	3.9 [kg]	4.9 [kg]
ours	**0.006**	**0.010**	**0.012**	**0.011**	**0.011**
ours w/o cond.	0.089	0.070	0.081	0.061	0.074

Table 7.5: Wasserstein distances between the acceleration distributions ("acc. dist") of ground-truth trajectory and the generated from *RatioNet* (Ours). We also show the same metric computed on the interpolated subdivided trajectory with an equal length.

	Ours	Interpolation
acc. dist. ↓	**0.379**	0.447

Eq. (7.20) are set to 10.0 and 1.0, respectively. For the fitting optimisation defined in Eq. (7.15), we set $\lambda_{\text{data}} = 1.0$, $\lambda_{touch} = 0.7$, $\lambda_{col.} = 0.8$ and $\lambda_{\text{prior}} = 0.001$. As suggested in Dabral et al. (2023), $\lambda_{\text{geo.}}$ of Eq. (7.10) and (7.20) are set such that larger penalties are applied with a smaller diffusion step t:

$$\lambda_{\text{geo.}} = \frac{10}{\exp \frac{10t}{T}}, \tag{7.27}$$

where T is the maximum diffusion step. We empirically set the maximum diffusion step T for *HandDiff* and *TrajDiff* to 150 and 300, respectively.

7.5.2 *Quantitative Results*

In this section, we evaluate the motion quality of *MACS* from various perspectives. We report a diversity and multi-modality measurement as suggested by Guo et al. (2020) in Table 7.1. We also evaluate the physical plausibility by measuring the following metrics:

Non-collision ratio (m_{col}) measures the ratio of frames with no hand-object collisions. A higher value indicates fewer collisions between the hand and the object.

Collision distance (m_{dist}) measures the distance of hand object penetration averaged over all the samples. A lower value indicates a lower magnitude of the collisions.

Table 7.6: Results of the user study (perceptual motion quality).

	GT	Ours	VAE	VAEGAN
reality score ↑	7.10±2.09	**6.01±2.08**	5.10±2.24	4.54±2.39

Non-touching ratio (m_{touch}) measures the ratio of samples over all the samples where there is no contact between the hand and object. A lower value indicates fewer *floating object* artefacts (*i.e.* spurious absence of contacts). Note that to report m_{touch}, we discard throwing motion action labels, as the assumption is that there should be constant contacts between the hands and the object. The hand vertices whose nearest distances to the object are lower than a threshold value of $5mm$ are considered contact vertices. Similarly, to compute m_{col} and m_{dist}, the interpenetrations over $5mm$ are considered collisions. To compute the metrics, we generate 500 samples across 6 different action labels.

DIVERSITY AND MULTIMODALITY Diversity measures the motion variance over all the frames within each action class, whereas multimodality measures the motion variance across the action classes. High diversity and multimodality indicate that the generated samples contain diversified motions. Please refer to Guo et al. (2020) for more details. We report the diversity and multimodality metrics for the generated hand motions and the object trajectories in Table 7.1. It is clear that in both cases *Ours* generates much more diversified motions when compared to the baselines, which we attribute to our diffusion model-based architecture. Notably, the generated trajectory samples contain more diversified motions compared with the metrics computed on the GT data.

PHYSICAL PLAUSIBILITY We report the physical plausibility measurements in Table 7.2. *Ours* shows the highest performance across all three metrics m_{col}, m_{dist} and m_{touch}. *VAE* yields m_{col} and m_{dist} comparable to *Ours*, however, its m_{touch} is substantially higher with 42% error increase compared to *Ours*. *VAEGAN* shows m_{touch} similar to *Ours* however it underperforms in terms of the collision-related metrics.

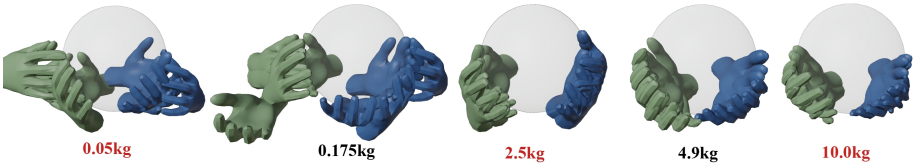


Figure 7.6: Grasp synthesis with different object masses. Our method can generate sequences influenced by masses close (in black) and far (in red) from the training dataset.

ABLATION STUDY Here, we motivate the use of the important loss terms of our fitting optimisation and training loss functions.

Interaction loss terms $\mathcal{L}_{touch}$ and $\mathcal{L}_{col.}$: In Table 7.2, we show the results of the fitting optimisation without $\mathcal{L}_{touch}$ and without $\mathcal{L}_{col.}$. When omitting the contact term $\mathcal{L}_{touch}$, the generated hands are not in contact with the object in most of the frames. This results in substantially higher metric m_{touch} and manifests through undesirable *floating object* artefacts. Omitting the collision term $\mathcal{L}_{col.}$ leads to frequent interpenetrations, lower m_{col} and higher m_{dist}. Therefore, it is essential to employ both the loss terms to generate sequences with higher physical plausibility.

Temporal loss terms $\mathcal{L}_{vel.}$ and $\mathcal{L}_{acc.}$: to report the ablative study of the loss terms $\mathcal{L}_{vel.}$ and $\mathcal{L}_{acc.}$ for the network training, we compute the Wasserstein distance between the accelerations of the sampled data and the GT data denoted as "acc. dist." in Table 7.3. Combining the two loss terms $\mathcal{L}_{vel.}$ and $\mathcal{L}_{acc.}$, our method shows the shortest distance from the GT acceleration distributions.

Plausibility of the conditioning mass value effect: can be evaluated by measuring the similarity between the GT object accelerations and the sampled ones. In Table 7.4, we show the "acc. dist." between the accelerations of the ground truth object motions and the sampled motions *with and without* mass conditioning. With the conditioning mass value, our network synthesises the motions with more physically plausible accelerations on each mass value compared with the network without a mass conditioning.

Effect of RatioNet on the user-provided trajectories: The goal of RatioNet is to provide plausible dynamics on the user-provided trajectories given conditioning mass values, *e.g.* higher object motion speed appears with lighter mass and the object is moved slower with heavier mass value.

For the ablative study of RatioNet, we report the "acc. dist." with and without RatioNet comparing with the acceleration distributions of our GT trajectories. For the component without RatioNet, we simply apply uniform sampling on the provided trajectories, denoted as "Interpolation" in Table 7.5. Thanks to our RatioNet, the object acceleration shows much more plausible values than without the network, faithfully responding to the conditioning mass values.

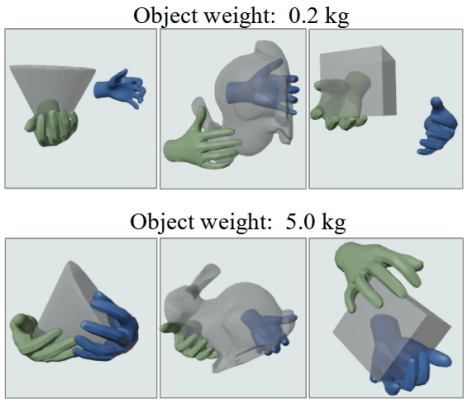


Figure 7.7: Example visualisations of 3D manipulations of the objects that are unseen during the network training, given conditioning mass values of 0.2kg (top row) and 5.0kg (bottom row).

USER STUDY The realism of 3D motions can be perceived differently depending on individuals. To quantitatively measure the plausibility of the synthesised motions, we perform an online user study. We prepared 26 questions with videos and gathered 42 participants in total. The videos for the study were randomly selected from the sampled results of VAE and VAEGAN baselines, *MACS* and the GT motions. In the first section, the subjects were asked to select the naturalness of the motions on a scale of 1 to 10 *reality score* (1 for completely unnatural and 10 for very natural). Table 7.6 shows the mean scores. *MACS* clearly outperforms other benchmarks in this perceptual user study, thanks to our diffusion-based networks that synthesise 3D manipulations with high-frequency details. In the additional section, we further evaluated our method regarding how faithfully the synthesised motions are affected by the conditional mass value. We show two videos of motions at once where the network is conditioned by mass values of 1.0 and 5.0, respectively. The participants were instructed to determine which sequence appeared to depict the manipulation of a heavier object. On average, the participants selected the correct answer with 92.8% accuracy, which suggests that *MACS* plausibly reflects the conditioning mass value in its motion.

7.5.3 *Qualitative Results*

INTERACTION SYNTHESIS In Fig. 7.1, we show the synthesised hand and object interaction sequence conditioned by the action labels and mass of the object. The synthesised motions show realistic and dynamic interactions between the hands and the object.

GRASP SYNTHESIS We show five samples of grasps for different conditioning mass values in Fig. 7.6. To generate this visualisation, we trained *HandDiff* without providing the action labels. In order to synthesise the graphs, we provide an object trajectory with position and rotations set to 0. Our method shows diverse grasps faithfully reflecting the conditional mass values. Most notably, the synthesised hands tend to support the heavy object at its bottom using the whole palm, whereas the light object tends to be supported using the fingertips only. Furthermore, the synthesised grasps show reasonable results even with unseen interpolated (2.5kg) and extrapolated (0.05kg and 10.0kg) mass values (highlighted in red).

UNSEEN OBJECTS We show the synthesised motions for objects that were not seen during the training, specifically a cone, the Stanford bunny and a cube in Fig. 7.7. Thanks to the synthesised hand contact labels conditioned by a mass value, *MACS* shows modest adaptations to different shapes while still correctly reflecting the provided mass values.

CONTACT VISUALISATION In Fig. 7.8 - (left), we provide visual examples of synthesised contacts with different mass values (0.18kg and 4.9kg). The synthesised contacts are distributed across the palm region when a heavier mass is given, whereas they concentrate around the fingertips with a lighter mass, which follows our intuition.

USER SPECIFIED TRAJECTORY Fig. 7.8 - (right) displays example synthesis results with user-provided input trajectories (S-curve and infinity curve). Thanks to the RatioNet, the object speed reflects the conditioning mass value, *i.e.* faster speed for lighter mass and vice versa.

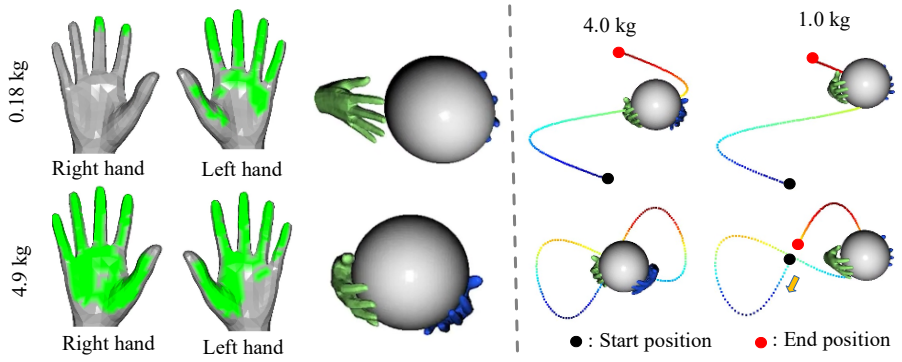


Figure 7.8: **(left)** Example visualisations of the contacts synthesised by *ConNet*, given conditioning mass values of 0.18 kg (top) and 4.9 kg (bottom). With heavier mass, the contact region spans the entire palm region, whereas contacts concentrate around the fingertips for a light object. **(right)** Example visualisations of 3D object manipulation given user input trajectories of S curve (top) and infinity curve (bottom). Thanks to the *RatioNet*, the object manipulation speed matches our intuition *i.e.* slower manipulation speed with heavier objects, and vice versa.

7.6 CONCLUSION

This chapter introduced the first approach to synthesise realistic 3D object manipulations with two hands faithfully responding to conditional mass. Our diffusion-model-based *MACS* approach produces plausible and diverse object manipulations, as verified quantitatively and qualitatively.

Since this topic has so far been completely neglected in the literature, the focus of this chapter is to demonstrate the impact of mass onto manipulation and hence we opted to use a single shape with uniform static mass distribution. As such there are several limitations that open up to exciting future work; for example the effect of shape diversity, non-uniform mass distribution (*i.e.* one side of the object is heavier than the other), or dynamic mass distribution (*e.g.* a bottle of water). Furthermore, we would like to highlight that other physical factors, such as friction or individual muscle strength, also impact object manipulation and could be addressed in future works. Lastly, while this chapter focused on synthesis with applications for ML data generation, entertainment and mixed reality experiences, we believe that weight analysis is another interesting avenue to explore, i.e. predicting the weight based on observed manipulation. This could be valuable in supervision scenarios to identify if an object changed its weight over time.

8

CONCLUSION

This thesis provided novel and various ways for capturing 3D human motion from a monocular view RGB video as well as 3D hand-object manipulation synthesis. The approaches introduce explicit physics and/or new modelling of human behaviour in daily life, such as environment interactions and body surface deformation for more realistic and accurate 3D reconstructions of human motions as well as the object manipulation synthesis explicitly conditioned by the object's mass.

Chapter 3 introduced a new monocular RGB-based human motion capture method by explicitly modelling the equations of motion, which runs in real time. The target kinematic pose and foot-floor contact information, as well as the motion state of the subject, are obtained from neural network based approaches that are utilised in the final physics-based motion tracking optimisation. Thanks to this explicit physics modelling in a motion estimation framework, the new approach exhibits much fewer artefacts (*e.g.*implausible collisions, joint jitters, foot skating and unnatural pose that is not achievable under gravity) than earlier purely kinematics-based approaches.

Despite the aforementioned advantages, there are a few limitations. For instance, this method uses a proportional derivative (PD) controller to control the humanoid character with physics quantities. The controller contains "fixed" coefficient values that adjust the intensity of the PD controller signal. Thus, the reconstructed 3D motions can show delayed motions when very fast motions, such as dancing, are given as inputs due to the static coefficient for the PD controllers. Chapter 4 tackled this issue by realising a fully learning-based approach with differentiable rigid body dynamics modelling. The networks estimate the coefficients of the PD controllers as well as the ground reaction forces to track even fast, challenging motions. The reconstructed 3D motions are plausible, handling foot-floor collisions. The approach estimates plausible forces and joint torques that can be visualised for applications like sports analysis, rehabilitation, etc.

This thesis also addressed MoCap with the simultaneous consideration of complex body-environment interaction in Chapter 5. The proposed method realises significantly improved 3D localisation over the prior works thanks to the guidance by the estimated body-environment contacts. Moreover, the novel sampling optimisation in a learned posed manifold space handles severe collisions in a hard manner. With those two novel components combined, the method is able to capture highly plausible human interactions in a complex scene from a monocular view input.

Chapter 6 proposed the first MoCap method that captures 3D hand and face motions along with the non-rigid facial skin deformations arising from their interactions. The new 3D deformation dataset was generated utilising a marker-less multi-view motion capture system combined with position based dynamics simulator. The proposed method trained on the dataset predicts face and hand motions along with non-rigid deformations of the face, showing highly natural self-interactions.

Chapter 7 proposed the first method to synthesise the 3D object manipulation with hands influenced by the conditioning object's mass value. The synthesised manipulations faithfully respond to the mass value. For instance, when dealing with a heavy object, the hands tend to support it from the bottom, utilising a large palm region, whereas a lighter object is often manipulated using the fingertips. Notably, the method optionally accepts as input the user-provided object trajectory and synthesises the natural manipulation of the object with plausible dynamics that follows the user-specified trajectory. This method represents the first of its kind and holds the potential to make significant contributions to various applications in machine learning and computer graphics.

8.1 INSIGHTS

In addition to the contributions introduced in this thesis for monocular RGB-based human motion capture and motion synthesis, this subsection provides the insights collected throughout these research works.

RESOLVING DEPTH AMBIGUITY This thesis proposed a series of monocular RGB-based MoCap methods while introducing several priors

to resolve the depth ambiguity of the single view setup. One of my observations is that estimating the human depth in a camera frame using a learning-based approach from single-view RGB inputs, without the awareness of focal length, can lead to severe overfitting to the training dataset. This occurs because the different combinations of focal lengths and subject depths can project the human figure to the same image plane location. Therefore, Chapter 4 first normalised the estimated 2D human key points using the known focal length before inputting them into the depth estimation network. This step abstracts away the focal length's impact from the input keypoints, which helped the network to generalise across different datasets that contain cameras with different focal lengths.

Chapter 6 also proposed a novel approach to reduce the depth ambiguity using the VAE-based learned interaction prior. This is based on the idea that the depth of the subject in the camera can differ significantly from sequence to sequence, the relative depths of the hands from the face do not. Therefore, the learned interaction prior is defined in the "canonical face frame", which resulted in a substantial improvement of the 3D localisation accuracy compared to the one without the prior. Utilising the contact information also immensely helps to disambiguate the depth. The collision signal between the environment and the human body indicates the inaccuracy of the relative depth. In case the estimated contacts are estimated, such information can also be utilised to reduce the depth ambiguity as demonstrated in Chapters 6 and 5.

EFFECTIVENESS OF EXPLICIT PHYSICS MODELLING This thesis demonstrated that incorporating physics-based modelling into a MoCap system significantly reduces visual artefacts. Such formulation is especially advantageous for reconstruction tasks when occlusions occur in the input view, as the physics modelling remains effective for such scenarios as well. However, it is important to note that purely kinematics-driven methods can outperform physics-based techniques regarding 3D joint position accuracy as observed in Chapters 3 and 4. This is because physics modelling serves as a motion regulariser, minimising visual artefacts at the possible expense of 3D precision. Therefore, evaluating both 3D positional error and motion plausibility together is important. Additionally, I emphasise the need for more detailed human body modelling for improved accuracy in the obtained 3D reconstruction. Our foot,

for instance, comprises over 20 intricately coordinated muscles enabling movement (*Muscles* 2018). However, current physics-based MoCap methods often employ very simple shape primitives for body representation (*e.g.* concatenation of a few cuboids to model a foot). Employing such a simplified body model while enforcing physics-based constraints in the methods can result in the over-regularisation of the reconstructed motion.

8.2 OUTLOOK

In this subsection, I provide the possible research directions of future work that can be derived from the research works introduced in this thesis.

AUTOREGRESSIVE MOTION SYNTHESIS CONSIDERING THE STATES Numerous techniques for human motion synthesis have been proposed in the literature, allowing users to control motion types through action labels, texts, and sound cues. This explicit controllability is invaluable for graphics applications. However, it comes at the cost of reduced motion variability. In the real world, our actions are often influenced by the environment and mind. For instance, in a warm room, someone might decide to shed a layer of clothing, or open a window if it is cooler outside. While we can often anticipate such real-world decisions, there is always an element of unpredictability. Leaving this *unpredictability* in motion synthesis can enhance the realism and immersion for downstream applications, such as the development of non-player characters (NPCs) in games and VR environments. Developing an auto-regressive motion synthesis method which responds to an individual's current state, such as their comfort level, can be a promising direction. This "comfort" metric can be influenced by environmental factors and the physical forces acting upon the individual, and the method could generate subsequent motions to mitigate any discomfort. Such an algorithm design paves the way for crafting more unpredictable, lifelike avatar behaviours.

INTERACTIONS WITH PHOTO REALISM In this thesis, the works primarily focusing on capturing or synthesising realistic motions with interactions were introduced. The downstream applications of these works

are AR/VR, mixed reality and avatar communication, which require not only motion realism but also photorealism. While several photorealistic implicit 3D representations exist, for instance, Neural radiance field (Mildenhall et al., 2020) and SDF+colour (Zhong et al., 2023), modelling physical interactions between the implicit scene and explicit representation (*e.g.* a hand mesh) is non-trivial. Recently, a new photorealistic scene representation, 3D Gaussian Splatting, was proposed (Kerbl et al., 2023). The 3D scene is represented as a set of 3D Gaussian with translation, rotation and colours, allowing to keep correspondence over time (Luiten et al., 2023). Unlike the existing photorealistic representations, this representation is explicit, making interaction modelling more intuitive. When paired with non-rigid deformation modellings, such as position based dynamics, interactions with the scene represented by 3D Gaussians can potentially create interactions of higher realism.

BIBLIOGRAPHY

A. Salem, Farhan and Ayman Aly (2015). "PD Controller Structures: Comparison and Selection for an Electromechanical System." In: *International Journal of Intelligent Systems and Applications (IJISA)* 7.2.

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

Achenbach, Jascha, Robert Brylka, Thomas Gietzen, Katja zum Hebel, Elmar Schömer, Ralf Schulze, Mario Botsch, and Ulrich Schwanecke (2018). "A multilinear model for bidirectional craniofacial reconstruction." In: *Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)*, pp. 67–76.

Adobe (2020). *Mixamo*. https://www.mixamo.com/. Accessed: 2020-04-15.

Agarap, Abien Fred (2018). "Deep learning using rectified linear units (relu)." In: *arXiv preprint arXiv:1803.08375*.

Agrawal, A., B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter (2019a). "Differentiable Convex Optimization Layers." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Agrawal, Akshay, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter (2019b). "Differentiable convex optimization layers." In: *Advances in neural information processing systems (NeurIPS)*.

Akada, Hiroyasu, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik (2022). "UnrealEgo: A New Dataset

for Robust Egocentric 3D Human Motion Capture." In: *European Conference on Computer Vision (ECCV)*.

Andrews, Sheldon, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell (2016). "Real-Time Physics-Based Motion Capture with Sparse Sensors." In: *European Conference on Visual Media Production (CVMP)*.

Andriluka, Mykhaylo, Leonid Pishchulin, Peter Gehler, and Bernt Schiele (2014). "Human Pose Estimation: New Benchmark and State of the Art Analysis." In: *Computer Vision and Pattern Recognition (CVPR)*.

Barzel, Ronen, John F. Hughes, and Daniel N. Wood (1996). "Plausible Motion Simulation for Computer Graphics Animation." In: *Eurographics Workshop on Computer Animation and Simulation (CAS)*.

Bergamin, Kevin, Simon Clavet, Daniel Holden, and James Richard Forbes (2019). "DReCon: data-driven responsive control of physics-based characters." In: *ACM Transactions on Graphics (TOG)* 38.6.

Bo, Liefeng and Cristian Sminchisescu (2008). "Twin Gaussian Processes for Structured Prediction." In: *International Journal of Computer Vision (IJCV)* 87, pp. 28–52.

Bogo, Federica, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black (2016). "Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image." In: *European Conference on Computer Vision (ECCV)*.

Bozic, Aljaz, Pablo Palafox, Michael Zollöfer, Angela Dai, Justus Thies, and Matthias Nießner (2020). "Neural Non-Rigid Tracking." In: *Advances in neural information processing systems (NeurIPS)*.

Bradley, Derek, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur (2008). "Markerless garment capture." In: *ACM Transactions on Graphics (TOG)* 27.3, p. 99.

Bradski, G. (2000). "The OpenCV Library." In: *Dr. Dobb's Journal of Software Tools*.

Brahmbhatt, Samarth, Cusuh Ham, Charles C Kemp, and James Hays (2019). "Contactdb: Analyzing and predicting grasp contact via thermal imaging." In: *Computer Vision and Pattern Recognition (CVPR)*.

Brau, Ernesto and Hao Jiang (2016). "3D Human Pose Estimation via Deep Learning from 2D Annotations." In: *International Conference on 3D Vision (3DV)*.

Brox, Thomas, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers (2010). "Combined Region and Motion-Based 3D Tracking of Rigid and Artic-

ulated Objects." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 32.3, pp. 402–415.

Bulat, Adrian and Georgios Tzimiropoulos (2017). "How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)." In: *International Conference on Computer Vision (ICCV)*.

Cagniart, Cedric, Edmond Boyer, and Slobodan Ilic (2010). "Free-Form Mesh Tracking: a Patch-Based Approach." In: *Computer Vision and Pattern Recognition (CVPR)*.

Cao, Zhe, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik (2020). "Long-term human motion prediction with scene context." In: *European Conference on Computer Vision (ECCV)*.

Cao, Zhe, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2019). "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 43.1, pp. 172–186.

Cao, Zhe, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik (2021). "Reconstructing hand-object interactions in the wild." In: *International Conference on Computer Vision (ICCV)*.

Cao, Zhe, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2017). "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields." In: *Computer Vision and Pattern Recognition (CVPR)*.

Casiez, Géry, Nicolas Roussel, and Daniel Vogel (2012). "1 € Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Charles, James, Tomas Pfister, Mark Everingham, and Andrew Zisserman (Oct. 2013). "Automatic and Efficient Human Pose Estimation for Sign Language Videos." In: *International Journal of Computer Vision (IJCV)* 110, pp. 70–90.

Chen, Ching-Hang and Deva Ramanan (2017). "3D Human Pose Estimation = 2D Pose Estimation + Matching." In: *Computer Vision and Pattern Recognition (CVPR)*.

Chentanez, Nuttapong, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke (2018). "Physics-based motion capture imitation with deep reinforcement learning." In: *Proceedings of the 11th*

Annual International Conference on Motion, Interaction, and Games, pp. 1–10.

Choi, Hongsuk, Gyeongsik Moon, and Kyoung Mu Lee (2021). "Beyond Static Features for Temporally Consistent 3D Human Pose and Shape from a Video." In: *Computer Vision and Pattern Recognition (CVPR)*.

Christen, Sammy, Muhammed Kocabas, Emre Aksan, Jemin Hwangbo, Jie Song, and Otmar Hilliges (2022). "D-Grasp: Physically Plausible Dynamic Grasp Synthesis for Hand-Object Interactions." In: *Computer Vision and Pattern Recognition (CVPR)*.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2015). "Fast and accurate deep network learning by exponential linear units (elus)." In: *arXiv preprint arXiv:1511.07289*.

Corona, Enric, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez (2020). "Ganhand: Predicting human grasp affordances in multi-object scenes." In: *Computer Vision and Pattern Recognition (CVPR)*.

Coros, Stelian, Philippe Beaudoin, and Michiel van de Panne (2010). "Generalized Biped Walking Control." In: *ACM Transactions on Graphics (TOG)* 29.4.

Coumans, Erwin and Yunfei Bai (2016). "Pybullet, a python module for physics simulation for games, robotics and machine learning." In: *GitHub repository*.

Dabral, Rishabh, Nitesh B Gundavarapu, Abhishek Mitra Rahuland Sharma, Ganesh Ramakrishnan, and Arjun Jain (2019). "Multi-Person 3D Human Pose Estimation from Monocular Images." In: *International Conference on 3D Vision (3DV)*.

Dabral, Rishabh, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt (2023). "MoFusion: A Framework for Denoising-Diffusion-based Motion Synthesis." In: *Computer Vision and Pattern Recognition (CVPR)*.

Dabral, Rishabh, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain (2018). "Learning 3D Human Pose from Structure and Motion." In: *European Conference on Computer Vision (ECCV)*.

Dabral, Rishabh, Soshi Shimada, Arjun Jain, Christian Theobalt, and Vladislav Golyanik (2021). "Gravity-Aware Monocular 3D Human-

Object Reconstruction." In: *International Conference on Computer Vision (ICCV)*.

Danecek, Radek, Michael J. Black, and Timo Bolkart (2022). "EMOCA: Emotion Driven Monocular Face Capture and Animation." In: *Computer Vision and Pattern Recognition (CVPR)*.

De Aguiar, Edilson, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun (2008). "Performance Capture from Sparse Multi-View Video." In: *ACM Transactions on Graphics (TOG)* 27.3.

Dejnabadi, Hooman, Brigitte M. Jolles, Emilio Casanova, Pascal Fua, and Kamiar Aminian (2006). "Estimation and visualization of sagittal kinematics of lower limbs orientation using body-fixed sensors." In: *Transactions on Biomedical Engineering (TBME)* 53.7, pp. 1385–1393.

Detry, Renaud, Dirk Kraft, Anders Glent Buch, Norbert Krüger, and Justus Piater (2010). "Refining grasp affordance models by experience." In: *International Conference on Robotics and Automation (ICRA)*.

Elhayek, Ahmed, Edilson de Aguiar, Arjun Jain, Jonathan Thompson, Leonid Pishchulin, Mykhaylo Andriluka, Christoph Bregler, Bernt Schiele, and Christian Theobalt (2016). "MARCOnI—ConvNet-Based MARker-Less Motion Capture in Outdoor and Indoor Scenes." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.3, pp. 501–514.

Elhayek, Ahmed, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Micha Andriluka, Chris Bregler, Bernt Schiele, and Christian Theobalt (2015). "Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras." In: *Computer Vision and Pattern Recognition (CVPR)*.

Elhayek, Ahmed, Carsten Stoll, Kwang In Kim, and Christian Theobalt (2014). "Outdoor Human Motion Capture by Simultaneous Optimization of Pose and Camera Parameters." In: *Computer Graphics Forum (CGF)*.

Fabbri, Matteo, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara (2020). "Compressed Volumetric Heatmaps for Multi-Person 3D Pose Estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Faloutsos, Petros, Michiel van de Panne, and Demetri Terzopoulos (2001). "Composable Controllers for Physics-Based Character Animation." In: *Computer Graphics and Interactive Techniques (CGIT)*.

Featherstone, Roy (2014). *Rigid body dynamics algorithms*.

Felis, Martin L. (2017). "RBDL: an Efficient Rigid-Body Dynamics Library using Recursive Algorithms." In: *Autonomous Robots* 41.2, pp. 495–511.

Feng, Yao, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black (2021a). "Collaborative Regression of Expressive Bodies using Moderation." In: *International Conference on 3D Vision (3DV)*.

Feng, Yao, Haiwen Feng, Michael J Black, and Timo Bolkart (2021b). "Learning an animatable detailed 3d face model from in-the-wild images." In: *ACM Transactions on Graphics (TOG)* 40.4.

Fieraru, Mihai, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu (2020). "Three-dimensional reconstruction of human interactions." In: *Computer Vision and Pattern Recognition (CVPR)*.

– (2021). "Learning complex 3D human self-contact." In: *AAAI Conference on Artificial Intelligence (AAAI)*.

Fuentes-Jimenez, David, Daniel Pizarro, David Casillas-Perez, Toby Collins, and Adrien Bartoli (2021). "Texture-Generic Deep Shape-From-Template." In: *IEEE Access* 9, pp. 75211–75230.

Gall, Juergen, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel (2010). "Optimization and Filtering for Human Motion Capture - a Multi-Layer Framework." In: *International Journal of Computer Vision (IJCV)* 87.1, pp. 75–92.

Gall, Juergen, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel (2009). "Motion Capture Using Joint Skeleton Tracking and Surface Estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Garrido, Pablo, Levi Valgaerts, Chenglei Wu, and Christian Theobalt (2013). "Reconstructing detailed dynamic face geometry from monocular video." In: *ACM Transactions on Graphics (TOG)* 32.6.

Garrido, Pablo, Michael Zollhöfer, Chenglei Wu, Derek Bradley, Patrick Pérez, Thabo Beeler, and Christian Theobalt (2016). "Corrective 3D reconstruction of lips from monocular video." In: *ACM Transactions on Graphics (TOG)* 35.6.

Gärtner, Erik, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu (2022a). "Differentiable dynamics for articulated 3d human motion reconstruction." In: *Computer Vision and Pattern Recognition (CVPR)*.

Gärtner, Erik, Mykhaylo Andriluka, Hongyi Xu, and Cristian Sminchisescu (2022b). "Trajectory optimization for physics-based reconstruction of 3d human pose from monocular video." In: *Computer Vision and Pattern Recognition (CVPR)*.

Ghosh, Anindita, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek (2023). "IMoS: Intent-Driven Full-Body Motion Synthesis for Human-Object Interactions." In: *Computer Graphics Forum (CGF)*.

Golyanik, Vladislav, Soshi Shimada, and Christian Theobalt (2020). "Fast simultaneous gravitational alignment of multiple point sets." In: *International Conference on 3D Vision (3DV)*.

Golyanik, Vladislav, Soshi Shimada, Kiran Varanasi, and Didier Stricker (2018). "Hdm-net: Monocular non-rigid 3d reconstruction with learned deformation model." In: *EuroVR International Conference*.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative Adversarial Nets." In: *Advances in Neural Information Processing Systems (NeurIPS)*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger.

Grady, Patrick, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp (2021). "Contactopt: Optimizing contact to improve grasps." In: *Computer Vision and Pattern Recognition (CVPR)*.

Guo, Chuan, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng (2020). "Action2motion: Conditioned generation of 3d human motions." In: *ACM International Conference on Multimedia (ICM)*.

Guo, Kaiwen, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu (2017). "Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera." In: *ACM Transactions on Graphics (TOG)* 36.4.

Habermann, Marc, Weipeng Xu, Helge Rhodin, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt (2018). "NRST: Non-rigid Sur-

face Tracking from Monocular Video." In: *German Conference on Pattern Recognition (GCPR)*.

Habermann, Marc, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt (2020). "DeepCap: Monocular Human Performance Capture Using Weak Supervision." In: *Computer Vision and Pattern Recognition (CVPR)*.

Habermann, Marc, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt (2019). "LiveCap: Real-Time Human Performance Capture From Monocular Video." In: *ACM Transactions on Graphics (TOG)* 38.2, 14:1–14:17.

Habibie, Ikhsanul, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt (2019). "In the Wild Human Pose Estimation Using Explicit 2D Features and Intermediate 3D Representations." In: *Computer Vision and Pattern Recognition (CVPR)*.

Hassan, Mohamed, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black (2021a). "Stochastic scene-aware motion prediction." In: *International Conference on Computer Vision (ICCV)*.

Hassan, Mohamed, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black (2019). "Resolving 3D Human Pose Ambiguities with 3D Scene Constraints." In: *International Conference on Computer Vision (ICCV)*.

Hassan, Mohamed, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black (2021b). "Populating 3D Scenes by Learning Human-Scene Interaction." In: *Computer Vision and Pattern Recognition (CVPR)*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition." In: *Computer Vision and Pattern Recognition (CVPR)*.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Hu, Haoyu, Xinyu Yi, Hao Zhang, Jun-Hai Yong, and Feng Xu (2022). "Physical Interaction: Reconstructing Hand-object Interactions with Physics." In: *SIGGRAPH Asia 2022 Conference Papers*.

Huang, Buzhen, Liang Pan, Yuan Yang, Jingyi Ju, and Yangang Wang (2022). "Neural MoCon: Neural Motion Control for Physically Plausible Human Motion Capture." In: *Computer Vision and Pattern Recognition (CVPR)*.

Ichim, Alexandru Eugen, Sofien Bouaziz, and Mark Pauly (2015). "Dynamic 3D avatar creation from hand-held video input." In: *ACM Transactions on Graphics (TOG)* 34.4.

Innmann, Matthias, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger (2016). "Volumedeform: Real-time volumetric non-rigid reconstruction." In: *International Conference on Computer Vision (ICCV)*.

Ionescu, Catalin, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu (2013). "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 36.7, pp. 1325–1339.

Jiang, Wen, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis (2020). "Coherent Reconstruction of Multiple Humans from a Single Image." In: *Computer Vision and Pattern Recognition (CVPR)*.

Jiang, Yifeng, Tom Van Wouwe, Friedl De Groote, and C. Karen Liu (2019). "Synthesis of Biologically Realistic Human Motion Using Joint Torque Actuation." In: *ACM Transactions on Graphics (TOG)* 38.4.

John, Vijay, Emanuele Trucco, and Stephen McKenna (2010). "Markerless human motion capture using charting and manifold constrained particle swarm optimisation." In: *British Machine Vision Conference (BMVC)*.

Johnson, Erik C.M., Marc Habermann, Soshi Shimada, Vladislav Golyanik, and Christian Theobalt (2023). "Unbiased 4D: Monocular 4D Reconstruction with a Neural Deformation Model." In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Johnson, S. and M. Everingham (2011). "Learning Effective Human Pose Estimation from Inaccurate Annotation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Kairanda, Navami, Edgar Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik (2022). "ϕ-SfT: Shape-from-Template with a Physics-based Deformation Model." In: *Computer Vision and Pattern Recognition (CVPR)*.

Kanazawa, Angjoo, Michael J. Black, David W. Jacobs, and Jitendra Malik (2018). "End-to-end Recovery of Human Shape and Pose." In: *Computer Vision and Pattern Recognition (CVPR)*.

Kanazawa, Angjoo, Jason Y. Zhang, Panna Felsen, and Jitendra Malik (2019). "Learning 3D Human Dynamics from Video." In: *Computer Vision and Pattern Recognition (CVPR)*.

Karunratanakul, Korrawe, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang (2020). "Grasping field: Learning implicit representations for human grasps." In: *International Conference on 3D Vision (3DV)*.

Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis (2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering." In: *ACM Transactions on Graphics (TOG)* 42.4.

Kingma, Diederik P. and Jimmy Ba (2015a). "Adam: A Method for Stochastic Optimization." In: *International Conference on Learning Representations, ICLR*. Ed. by Yoshua Bengio and Yann LeCun.

Kingma, Diederik P and Jimmy Ba (2015b). "Adam: A method for stochastic optimization." In: *International Conference on Learning Representations (ICLR)*.

Kingma, Diederik P and Max Welling (2014). "Auto-encoding variational bayes." In: *International Conference on Learning Representations (ICLR)*.

Knauer, Christian, Maarten Löffler, Marc Scherfenberg, and Thomas Wolle (2009). "The Directed Hausdorff Distance between Imprecise Point Sets." In: *International Symposium on Algorithms and Computation (ISAAC)*.

Kocabas, Muhammed, Nikos Athanasiou, and Michael J. Black (2020a). "VIBE: Video Inference for Human Body Pose and Shape Estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Kocabas, Muhammed, Nikos Athanasiou, and Michael J Black (2020b). "VIBE: Video inference for human body pose and shape estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Kocabas, Muhammed, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black (2021a). "PARE: Part Attention Regressor for 3D Human Body Estimation." In: *International Conference on Computer Vision (ICCV)*.

Kocabas, Muhammed, Chun-Hao P. Huang, Joachim Tesch, Lea Müller, Otmar Hilliges, and Michael J. Black (2021b). "SPEC: Seeing People in the Wild with an Estimated Camera." In: *International Conference on Computer Vision (ICCV)*.

Kolotouros, Nikos, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis (2019). "Learning to Reconstruct 3D Human Pose and Shape

via Model-fitting in the Loop." In: *International Conference on Computer Vision (ICCV)*.

Kolotouros, Nikos, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis (2021). "Probabilistic Modeling for Human Mesh Recovery." In: *International Conference on Computer Vision (ICCV)*.

Kong, Zhifeng, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro (2021). "Diffwave: A versatile diffusion model for audio synthesis." In: *International Conference on Learning Representations (ICLR)*.

Kovalenko, Onorina, Vladislav Golyanik, Jameel Malik, Ahmed Elhayek, and Didier Stricker (2019). "Structure from Articulated Motion: Accurate and Stable Monocular 3D Reconstruction without Training Data." In: *Sensors* 19.20.

Krug, Robert, Dimitar Dimitrov, Krzysztof Charusta, and Boyko Iliev (2010). "On the efficient computation of independent contact regions for force closure grasps." In: *International Conference on Intelligent Robots and Systems (ICIRS)*.

Kwok, Yen Lee Angela, Jan Gralton, and Mary-Louise McLaws (2015). "Face touching: a frequent habit that has implications for hand hygiene." In: *American journal of infection control (AJIC)* 43.2, pp. 112–114.

Lattas, Alexandros, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou (2020). "AvatarMe: Realistically Renderable 3D Facial Reconstruction" in-the-wild"." In: *Computer Vision and Pattern Recognition (CVPR)*.

Lee, Seunghwan, Moonseok Park, Kyoungmin Lee, and Jehee Lee (2019). "Scalable Muscle-Actuated Human Simulation and Control." In: *ACM Transactions on Graphics (TOG)* 38.4.

Levenberg, Kenneth (1944). "A method for the solution of certain non-linear problems in least squares." In: *Quarterly Journal of Applied Mathmatics (AMS)* II.2, pp. 164–168.

Levine, Sergey and Jovan Popović (2012). "Physically Plausible Simulation for Character Animation." In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*.

Li, Tianye, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero (2017). "Learning a model of facial shape and expression from 4D scans." In: *ACM Transactions on Graphics (TOG)* 36.6, 194:1–194:17.

Li, Ying, Jiaxin L Fu, and Nancy S Pollard (2007). "Data-driven grasp synthesis using shape matching and task-based pruning." In: *Transactions on visualization and computer graphics (TVCG)* 13.4, pp. 732–747.

Li, Zhi, Soshi Shimada, Bernt Schiele, Christian Theobalt, and Vladislav Golyanik (2022). "MoCapDeform: Monocular 3D Human Motion Capture in Deformable Scenes." In: *International Conference on 3D Vision (3DV)*.

Li, Zongmian, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic (2019). "Estimating 3D Motion and Forces of Person-Object Interactions from Monocular Video." In: *Computer Vision and Pattern Recognition (CVPR)*.

Lin, Wenbin, Chengwei Zheng, Jun-Hai Yong, and Feng Xu (2022). "Occlusionfusion: Occlusion-aware motion estimation for real-time dynamic 3d reconstruction." In: *Computer Vision and Pattern Recognition (CVPR)*.

Liu, Libin, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu (2010). "Sampling-Based Contact-Rich Motion Control." In: *ACM Transactions on Graphics (TOG)* 29.4, 128:1–128:10.

Liu, Shaowei, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang (2021). "Semi-supervised 3d hand-object poses estimation with interactions in time." In: *Computer Vision and Pattern Recognition (CVPR)*.

Liu, Yebin, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt (2011). "Markerless Motion Capture of Interacting Characters using Multi-View Image Segmentation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, et al. (2019). "Mediapipe: A framework for perceiving and processing reality." In: *Workshop on Computer Vision for AR/VR at Computer Vision and Pattern Recognition (CVPRW)*.

Luiten, Jonathon, Georgios Kopanas, Bastian Leibe, and Deva Ramanan (2023). "Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis." In: *arXiv preprint arXiv:2308.09713*.

Luo, Zhengyi, Ryo Hachiuma, Ye Yuan, and Kris Kitani (2021). "Dynamics-regulated kinematic policy for egocentric pose estimation." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Luo, Zhengyi, Shun Iwase, Ye Yuan, and Kris Kitani (2022). "Embodied Scene-aware Human Pose Estimation." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Maas, Andrew L, Awni Y Hannun, Andrew Y Ng, et al. (2013). "Rectifier nonlinearities improve neural network acoustic models." In: *International Conference on Machine Learning (ICML)*.

Macchietto, Adriano, Victor Zordan, and Christian R. Shelton (2009). "Momentum Control for Balance." In: *ACM Transactions on Graphics (TOG)* 28.3.

Mahmood, Naureen, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black (2019). "AMASS: Archive of Motion Capture as Surface Shapes." In: *International Conference on Computer Vision (ICCV)*.

Malik, Jameel, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker (2020). "Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map." In: *Computer Vision and Pattern Recognition (CVPR)*.

Malik, Jameel, Soshi Shimada, Ahmed Elhayek, Sk Aziz Ali, Christian Theobalt, Vladislav Golyanik, and Didier Stricker (2021). "Handvoxnet++: 3d hand shape and pose estimation using voxel-based neural networks." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 44.12, pp. 8962–8974.

Marcard, Timo, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll (2017). "Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs." In: 36.2, pp. 349–360.

Marquardt, Donald W. (1963). "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." In: *SIAM Journal on Applied Mathematics (SIAP)* 11.2, pp. 431–441.

Martin-Brualla, Ricardo, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello (2018). "LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering." In: *ACM Transactions on Graphics (TOG)* 37.6.

Martinez, Julieta, Rayat Hossain, Javier Romero, and James J. Little (2017). "A Simple Yet Effective Baseline for 3D Human Pose Estimation." In: *International Conference on Computer Vision (ICCV)*.

Mehta, Dushyant, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt (2017a). "Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision." In: *International Conference on 3D Vision (3DV)*.

Mehta, Dushyant, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohammad Elgharib, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt (2020). "XNect: Real-time Multi-Person 3D Motion Capture with a Single RGB Camera." In: *ACM Transactions on Graphics (TOG)* 39.4.

Mehta, Dushyant, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt (2017b). "VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera." In: vol. 36. 4.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In: *European Conference on Computer Vision (ECCV)*.

Monszpart, Aron, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J. Mitra (2019). "IMapper: Interaction-Guided Scene Mapping from Monocular Videos." In: *ACM Transactions on Graphics (TOG)* 38.4.

Mordatch, Igor, Zoran Popović, and Emanuel Todorov (2012). "Contact-invariant optimization for hand manipulation." In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*.

Moreno-Noguer, Francesc (2017). "3D Human Pose Estimation From a Single Image via Distance Matrix Regression." In: *Computer Vision and Pattern Recognition (CVPR)*.

Mueller, Franziska, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt (2019). "Real-time pose and shape reconstruction of two interacting hands with a single depth camera." In: *ACM Transactions on Graphics (TOG)* 38.4.

Müller, Lea, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black (2021). "On Self-Contact and Human Pose." In: *Computer Vision and Pattern Recognition (CVPR)*.

Müller, Matthias, Bruno Heidelberger, Marcus Hennix, and John Ratcliff (2007). "Position based dynamics." In: *Visual Communication and Image Representation (VCIR)* 18.2, pp. 109–118.

Muscles (2018). https://www.healthline.com/human-body-maps/foot-muscles. Accessed: 2023-10-31.

Nakaoka, Shin'ichiro, Atsushi Nakazawa, Fumio Kanehiro, Kenji Kaneko, Mitsuharu Morisawa, Hirohisa Hirukawa, and Katsushi Ikeuchi (2007). "Learning from Observation Paradigm: Leg Task Models for Enabling a Biped Humanoid Robot to Imitate Human Dances." In: *International Journal of Robotics Research (IJRR)* 26.8, pp. 829–844.

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). "Stacked Hourglass Networks for Human Pose Estimation." In: *European Conference on Computer Vision (ECCV)*.

Ngo, Dat Tien, Sanghyuk Park, Anne Jorstad, Alberto Crivellaro, Chang D. Yoo, and Pascal Fua (2015). "Dense Image Registration and Deformable Surface Reconstruction in Presence of Occlusions and Minimal Texture." In: *International Conference on Computer Vision (ICCV)*.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. (2019). "Pytorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Pavlakos, Georgios, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black (2019). "Expressive Body Capture: 3D Hands, Face, and Body from a Single Image." In: *Computer Vision and Pattern Recognition (CVPR)*.

Pavlakos, Georgios, Xiaowei Zhou, and Kostas Daniilidis (2018a). "Ordinal Depth Supervision for 3D Human Pose Estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Pavlakos, Georgios, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis (2017). "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose." In: *Computer Vision and Pattern Recognition (CVPR)*.

Pavlakos, Georgios, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis (2018b). "Learning to estimate 3D human pose and shape from a single color image." In: *Computer Vision and Pattern Recognition (CVPR)*.

Pavllo, Dario, Christoph Feichtenhofer, David Grangier, and Michael Auli (2019). "3d human pose estimation in video with temporal convolutions and semi-supervised training." In: *Computer Vision and Pattern Recognition (CVPR)*.

Peng, Xue Bin, Pieter Abbeel, Sergey Levine, and Michiel van de Panne (2018a). "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills." In: *ACM Transactions on Graphics (TOG)* 37.4.

Peng, Xue Bin, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine (2018b). "SFV: Reinforcement Learning of Physical Skills from Videos." In: *ACM Transactions on Graphics (TOG)* 37.6.

Petit, Antoine, Stéphane Cotin, Vincenzo Lippiello, and Bruno Siciliano (2018). "Capturing deformations of interacting non-rigid objects using rgb-d data." In: *International Conference on Intelligent Robots and Systems (IROS)*.

Pollard, Nancy S and Victor Brian Zordan (2005). "Physically based grasping control from example." In: *ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA)*.

Poole, Ben, Ajay Jain, Jonathan T Barron, and Ben Mildenhall (2023). "Dreamfusion: Text-to-3d using 2d diffusion." In: *International Conference on Learning Representations (ICLR)*.

Putri, Dewi Indriati Hadi, Carmadi Machbub, et al. (2018). "Gait Controllers on Humanoid Robot Using Kalman Filter and PD Controller." In: *International Conference on Control, Automation, Robotics and Vision (ICARCV)*.

Rempe, Davis, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas (2021). "HuMoR: 3D Human Motion Model for Robust Pose Estimation." In: *International Conference on Computer Vision (ICCV)*.

Rempe, Davis, Leonidas J Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang (2020). "Contact and Human Dynamics from Monocular Video." In: *European Conference on Computer Vision (ECCV)*.

Rhodin, Helge, Mathieu Salzmann, and Pascal Fua (2018). "Unsupervised Geometry-Aware Representation Learning for 3D Human Pose Estimation." In: *European Conference on Computer Vision (ECCV)*.

Rogez, Grégory, Philippe Weinzaepfel, and Cordelia Schmid (2019). "LCR-Net++: Multi-Person 2D and 3D Pose Detection in Natural Images." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 42.5, pp. 1146–1161.

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer (2022). "High-resolution image synthesis with latent diffusion models." In: *Computer Vision and Pattern Recognition (CVPR)*.

Romero, Javier, Dimitrios Tzionas, and Michael J. Black (Nov. 2017). "Embodied Hands: Modeling and Capturing Hands and Bodies Together." In: *ACM Transactions on Graphics (TOG)*. 245:1–245:17 36.6.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation." In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. (2022). "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding." In: *Advances in neural information processing systems (NeurIPS)*.

Saini, Sanjay, Dayang Rohaya Bt Awang Rambli, Suziah Bt Sulaiman, and M Nordin B Zakaria (2013). "Human pose tracking in low-dimensional subspace using manifold learning by charting." In: *International Conference on Signal and Image Processing Applications (ICSIPA)*.

Saini, Sanjay, Dayang Rohaya Bt Awang Rambli, Suziah Bt Sulaiman, M Nordin B Zakaria, and Siti Rohkmah (2012). "Markerless multi-view human motion tracking using manifold model learning by charting." In: *Procedia Engineering* 41, pp. 664–670.

Saito, Shunsuke, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li (2019). "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization." In: *International Conference on Computer Vision (ICCV)*.

Saito, Shunsuke, Tianye Li, and Hao Li (2016). "Real-time facial segmentation and performance capture from rgb input." In: *European Conference on Computer Vision (ECCV)*.

Salzmann, Mathieu, Julien Pilet, Slobodan Ilic, and Pascal Fua (2007). "Surface Deformation Models for Nonrigid 3D Shape Recovery." In: *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 29.8, pp. 1481–1487.

Schroder, Matthias and Helge Ritter (2017). "Hand-object interaction detection with fully convolutional networks." In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Sengupta, Soumyadip, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman (2020). "Background Matting: The World is Your Green Screen." In: *Computer Vision and Pattern Regognition (CVPR)*.

Shahabpoor, Erfan and Aleksandar Pavic (2017). "Measurement of Walking Ground Reactions in Real-Life Environments: A Systematic Review of Techniques and Technologies." In: *Sensors* 17.9, p. 2085.

Sharma, Saurabh, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain (2019). "Monocular 3D Human Pose Estimation by Generation and Ordinal Ranking." In: *International Conference on Computer Vision (ICCV)*.

Sharon, Dana and Michiel van de Panne (2005). "Synthesis of Controllers for Stylized Planar Bipedal Walking." In: *International Conference on Robotics and Animation (ICRA)*.

Shi, Mingyi, Kfir Aberman, Andreas Aristidou, Taku Komura, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen (2020). "MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency." In: *ACM Transactions on Graphics (TOG)* 40.1, pp. 1–15.

Shimada, Soshi, Vladislav Golyanik, Zhi Li, Patrick Pérez, Weipeng Xu, and Christian Theobalt (2022). "HULC: 3D HUman Motion Capture with Pose Manifold SampLing and Dense Contact Guidance." In: *European Conference on Computer Vision (ECCV)*.

Shimada, Soshi, Vladislav Golyanik, Patrick Pérez, and Christian Theobalt (2023). "Decaf: Monocular Deformation Capture for Face and Hand Interactions." In: *ACM Transactions on Graphics (TOG)* 42.6.

Shimada, Soshi, Vladislav Golyanik, Christian Theobalt, and Didier Stricker (2019). "Ismo-gan: Adversarial learning for monocular non-rigid 3d reconstruction." In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Shimada, Soshi, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt (2021). "Neural Monocular 3D Human Motion Capture with Physical Awareness." In: *ACM Transactions on Graphics (TOG)* 40.4.

Shimada, Soshi, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt (2020). "PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time." In: *ACM Transactions on Graphics (TOG)* 39.6.

Shimada, Soshi, Franziska Mueller, Jan Bednarik, Bardia Doosti, Bernd Bickel, Danhang Tang, Vladislav Golyanik, Jonathan Taylor, Christian Theobalt, and Thabo Beeler (2024). "MACS: Mass Conditioned 3D Hand and Object Motion Synthesis." In: *International Conference on 3D Vision (3DV)*.

Simon, Tomas, Hanbyul Joo, Iain Matthews, and Yaser Sheikh (2017). "Hand Keypoint Detection in Single Images using Multiview Bootstrapping." In: *Computer Vision and Pattern Recognition (CVPR)*.

Slavcheva, Miroslava, Maximilian Baust, Daniel Cremers, and Slobodan Ilic (2017). "Killingfusion: Non-rigid 3d reconstruction without correspondences." In: *Computer Vision and Pattern Recognition (CVPR)*.

Sohl-Dickstein, Jascha, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli (2015). "Deep unsupervised learning using nonequilibrium thermodynamics." In: *International Conference on Machine Learning (ICML)*.

Sohn, Kihyuk, Honglak Lee, and Xinchen Yan (2015). "Learning structured output representation using deep conditional generative models." In: *Advances in neural information processing systems (NeurIPS)*.

Song, Jie, Xu Chen, and Otmar Hilliges (2020). "Human Body Model Fitting by Learned Gradient Descent." In: *European Conference on Computer Vision (ECCV)*.

Starck, Jonathan and Adrian Hilton (2007). "Surface capture for performance-based animation." In: *Computer Graphics and Applications (CGA)* 27.3, pp. 21–31.

Stoll, Carsten, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt (2011). "Fast articulated motion tracking using a sums of Gaussians body model." In: *International Conference on Computer Vision (ICCV)*.

Sugihara, Tomomichi and Yoshihiko Nakamura (2006). "Gravity compensation on humanoid robot control with robust joint servo and non-integrated rate-gyroscope." In: *International Conference on Humanoid Robots (ICHR)*.

Sun, Yu, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei (2021). "Monocular, one-stage, regression of multiple 3d people." In: *International Conference on Computer Vision (ICCV)*.

Sun, Yu, Yun Ye, Wu Liu, Wenpeng Gao, YiLi Fu, and Tao Mei (2019). "Human mesh recovery from monocular images via a skeleton-disentangled representation." In: *International Conference on Computer Vision (ICCV)*.

Taheri, Omid, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas (2022). "GOAL: Generating 4D Whole-Body Motion for Hand-Object Grasping." In: *Computer Vision and Pattern Recognition (CVPR)*.

Taheri, Omid, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas (2020). "GRAB: A Dataset of Whole-Body Human Grasping of Objects." In: *European Conference on Computer Vision (ECCV)*.

Tautges, Jochen, Arno Zinke, Björn Krüger, Jan Baumann, Andreas Weber, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bernd Eberhardt (2011). "Motion Reconstruction Using Sparse Accelerometer Data." In: *ACM Transactions on Graphics (TOG)* 30.3.

Tekin, Bugra, Federica Bogo, and Marc Pollefeys (2019). "H+ o: Unified egocentric recognition of 3d hand-object poses and interactions." In: *Computer Vision and Pattern Recognition (CVPR)*.

Tekin, Bugra, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua (2016). "Structured Prediction of 3D Human Pose with Deep Neural Networks." In: *British Machine Vision Conference (BMVC)*.

Tevet, Guy, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano (2023). "Human Motion Diffusion Model." In: *International Conference on Learning Representations (ICLR)*.

Tewari, Ayush, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian (2017). "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction." In: *International Conference on Computer Vision (ICCV)*.

The Captury (2023). https://captury.com/.

Thies, Justus, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner (2016). "Face2face: Real-time face capture and reenactment of rgb videos." In: *Computer Vision and Pattern Recognition (CVPR)*.

Thobbi, Anand and Weihua Sheng (2010). "Imitation learning of hand gestures and its evaluation for humanoid robots." In: *International Conference on Information and Automation (ICIA)*.

Tomè, Denis, Chris Russell, and Lourdes Agapito (2017). "Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image." In: *Computer Vision and Pattern Recognition (CVPR)*.

Tretschk, Edith, Navami Kairanda, Mallikarjun BR, Rishabh Dabral, Adam Kortylewski, Bernhard Egger, Marc Habermann, Pascal Fua, Christian Theobalt, and Vladislav Golyanik (2023). "State of the Art in Dense Monocular Non-Rigid 3D Reconstruction." In: *Computer Graphics Forum (CGF)*. Vol. 42. 2, pp. 485–520.

Tsoli, Aggeliki and Antonis A Argyros (2018). "Joint 3D tracking of a deformable object in interaction with a hand." In: *European Conference on Computer Vision (ECCV)*.

Vicon blade (n.d.). https://www.vicon.com/.

Vlasic, Daniel, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović (2007). "Practical Motion Capture in Everyday Surroundings." In: *ACM Transactions on Graphics (TOG)* 26.3.

Vlasic, Daniel, Ilya Baran, Wojciech Matusik, and Jovan Popović (2008). "Articulated mesh animation from multi-view silhouettes." In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3, p. 97.

Vlasic, Daniel, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik (2009). "Dynamic Shape Capture using Multi-View Photometric Stereo." In: *ACM Transactions on Graphics (TOG)* 28.5, p. 174.

Vondrak, Marek, Leonid Sigal, Jessica Hodgins, and Odest Jenkins (2012). "Video-based 3D Motion Capture Through Biped Control." In: *ACM Transactions on Graphics (TOG)* 31.4, pp. 1–12.

Wandt, Bastian and Bodo Rosenhahn (2019). "RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Wang, Jiashun, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang (2021). "Synthesizing long-term 3d human motion and interaction in 3d scenes." In: *Computer Vision and Pattern Recognition (CVPR)*.

Wang, Jiayi, Diogo Luvizon, Franziska Mueller, Florian Bernard, Adam Kortylewski, Dan Casas, and Christian Theobalt (2022a). "HandFlow:

Quantifying View-Dependent 3D Ambiguity in Two-Hand Reconstruction with Normalizing Flow." In: *Vision, Modeling, and Visualization (VMV)*.

Wang, Jiayi, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A Otaduy, Dan Casas, and Christian Theobalt (2020a). "Rgb2hands: real-time tracking of 3d hand interactions from monocular rgb video." In: *ACM Transactions on Graphics (TOG)* 39.6.

Wang, Yangang, Yebin Liu, Xin Tong, Qionghai Dai, and Ping Tan (2018). "Robust Non-rigid Motion Tracking and Surface Reconstruction Using L0 Regularization." In: *Transctions on Visualization and Computer Graphics (TVCG)* 24.5, pp. 1770–1783.

Wang, Zhe, Liyan Chen, Shauray Rathore, Daeyun Shin, and Charless Fowlkes (2022b). "Geometric Pose Affordance: 3D Human Pose with Scene Constraints." In: *European Conference on Computer Vision Workshop (ECCVW)*.

Wang, Zhe, Daeyun Shin, and Charless Fowlkes (2020b). "Predicting Camera Viewpoint Improves Cross-dataset Generalization for 3D Human Pose Estimation." In: *European Conference on Computer Vision Workshop (ECCVW)*.

Waschbüsch, Michael, Stephan Würmlin, Daniel Cotting, Filip Sadlo, and Markus Gross (2005). "Scalable 3D Video of Dynamic Scenes." In: *The Visual Computer* 21.8-10, pp. 629–638.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional pose machines." In: *Computer Vision and Pattern Recognition (CVPR)*.

Wei, Xiaolin and Jinxiang Chai (2010). "Videomocap: Modeling Physically Realistic Human Motion from Monocular Video Sequences." In: *ACM Transactions on Graphics (TOG)*. Vol. 29. 4.

Winkler, Alexander W., C. Dario Bellicoso, Marco Hutter, and Jonas Buchli (2018). "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization." In: *Robotics and Automation Letters (RAL)* 3.3, pp. 1560–1567.

Wrotek, Pawel, Odest Chadwicke Jenkins, and Morgan McGuire (2006). "Dynamo: Dynamic, Data-Driven Character Control with Adjustable Balance." In: *ACM Sandbox Symposium on Video Games 2006*.

Wu, Chenglei, Derek Bradley, Markus Gross, and Thabo Beeler (2016). "An anatomically-constrained local deformation model for monocular face capture." In: *ACM Transactions on Graphics (TOG)* 35.4, pp. 1–12.

Wu, Chenglei, Kiran Varanasi, and Christian Theobalt (2012). "Full Body Performance Capture under Uncontrolled and Varying Illumination: A Shading-Based Approach." In: *European Conference on Computer Vision (ECCV)*.

Xiang, Donglai, Hanbyul Joo, and Yaser Sheikh (2019). "Monocular Total Capture: Posing Face, Body, and Hands in the Wild." In: *Computer Vision and Pattern Recognition (CVPR)*.

Xie, Haoran, Atsushi Watatani, and Kazunori Miyata (2019). "Visual feedback for core training with 3d human shape and pose." In: *Nicograph International (NicoInt)*.

Xie, Kevin, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti (2021). "Physics-based human motion estimation and synthesis from videos." In: *International Conference on Computer Vision (ICCV)*.

Xu, Hongyi, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu (2020). "Ghum & ghuml: Generative 3d human shape and articulated pose models." In: *Computer Vision and Pattern Recognition (CVPR)*.

Xu, Lan, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt (2020). "EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera." In: *Computer Vision and Pattern Recognition (CVPR)*.

Xu, Weipeng, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt (2018). "MonoPerfCap: Human Performance Capture From Monocular Video." In: *ACM Transactions on Graphics (TOG)* 37.2.

Yang, Chifu, Qitao Huang, Hongzhou Jiang, O Ogbobe Peter, and Junwei Han (2010). "PD control with gravity compensation for hydraulic 6-DOF parallel manipulator." In: *Mechanism and Machine Theory (MMT)* 45.4, pp. 666–677.

Yang, Wei, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang (2018). "3D Human Pose Estimation in the Wild by Adversarial Learning." In: *Computer Vision and Pattern Recognition (CVPR)*.

Ye, Yuting and C Karen Liu (2012). "Synthesis of detailed hand manipulations using contact sampling." In: *ACM Transactions on Graphics (TOG)* 31.4.

Yi, Xinyu, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu (2022). "Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors." In: *Computer Vision and Pattern Recognition (CVPR)*.

Yu, Alex (2023). *Triangle mesh to signed-distance function (SDF)*. https://github.com/sxyu/sdf.

Yu, Rui, Chris Russell, Neill DF Campbell, and Lourdes Agapito (2015). "Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video." In: *International Conference on Computer Vision (ICCV)*.

Yu, Xianwen, Xiaoning Zhang, Yang Cao, and Min Xia (2019). "VAEGAN: A Collaborative Filtering Framework based on Adversarial Variational Autoencoders." In: *International Joint Conference on Artificial Intelligence (IJCAI)*.

Yuan, Ye and Kris Kitani (2020). "Residual Force Control for Agile Human Behavior Imitation and Extended Motion Synthesis." In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuan, Ye, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz (2023). "PhysDiff: Physics-Guided Human Motion Diffusion Model." In: *International Conference on Computer Vision (ICCV)*.

Yuan, Ye, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih (2021). "Simpoe: Simulated character control for 3d human pose estimation." In: *Computer vision and pattern recognition (CVPR)*.

Zanfir, Andrei, Elisabeta Marinoiu, and Cristian Sminchisescu (2018). "Monocular 3D Pose and Shape Estimation of Multiple People in Natural Scenes - The Importance of Multiple Scene Constraints." In: *Computer Vision and Pattern Recognition (CVPR)*.

Zell, Petrissa, Bodo Rosenhahn, and Bastian Wandt (2020). "Weakly-Supervised Learning of Human Dynamics." In: *European Conference on Computer Vision (ECCV)*.

Zell, Petrissa, Bastian Wandt, and Bodo Rosenhahn (2017). "Joint 3D Human Motion Capture and Physical Analysis from Monocular Videos." In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Zhang, Baowen, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang (2021a). "Interacting two-hand 3d pose and shape reconstruction from single color image." In: *International Conference on Computer Vision (ICCV)*.

Zhang, Hao, Zi-Hao Bo, Jun-Hai Yong, and Feng Xu (2019). "InteractionFusion: real-time reconstruction of hand poses and deformable objects in hand-object interactions." In: *ACM Transactions on Graphics (TOG)* 38.4.

Zhang, Hao, Yuxiao Zhou, Yifei Tian, Jun-Hai Yong, and Feng Xu (2021b). "Single depth view based real-time reconstruction of hand-object interactions." In: *ACM Transactions on Graphics (TOG)* 40.3.

Zhang, He, Yuting Ye, Takaaki Shiratori, and Taku Komura (2021c). "ManipNet: Neural manipulation synthesis with a hand-object spatial representation." In: *ACM Transactions on Graphics (TOG)* 40.4, pp. 1–14.

Zhang, Jason Y., Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa (2020a). "Perceiving 3D Human-Object Spatial Arrangements from a Single Image in the Wild." In: *European Conference on Computer Vision (ECCV)*.

Zhang, Mingyuan, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu (2022). "Motiondiffuse: Text-driven human motion generation with diffusion model." In: *arXiv preprint arXiv:2208.15001*.

Zhang, Peizhao, Kristin Siu, Jianjie Zhang, C Karen Liu, and Jinxiang Chai (2014). "Leveraging Depth Cameras and Wearable Pressure Sensors for Full-Body Kinematics and Dynamics Capture." In: *ACM Transactions on Graphics (TOG)* 33.6, pp. 1–14.

Zhang, Siwei, Yan Zhang, Federica Bogo, Pollefeys Marc, and Siyu Tang (Oct. 2021d). "Learning Motion Priors for 4D Human Body Capture in 3D Scenes." In: *International Conference on Computer Vision (ICCV)*.

Zhang, Tianshu, Buzhen Huang, and Yangang Wang (2020b). "Object-Occluded Human Shape and Pose Estimation From a Single Color Image." In: *Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Yuxiang, Liang An, Tao Yu, xiu Li, Kun Li, and Yebin Liu (2020c). "4D Association Graph for Realtime Multi-Person Motion Capture Using Multiple Video Cameras." In: *International Conference on Computer Vision (ICCV)*.

Zheng, Juntian, Qingyuan Zheng, Lixing Fang, Yun Liu, and Li Yi (2023). "CAMS: CAnonicalized Manipulation Spaces for Category-Level Functional Hand-Object Manipulation Synthesis." In: *Computer Vision and Pattern Recognition (CVPR)*.

Zheng, Yu and Katsu Yamane (2013). "Human Motion Tracking Control with Strict Contact Force Constraints for Floating-Base Humanoid Robots." In: *International Conference on Humanoid Robots (Humanoids)*.

Zhong, Licheng, Lixin Yang, Kailin Li, Haoyu Zhen, Mei Han, and Cewu Lu (2023). "Color-NeuS: Reconstructing Neural Implicit Surfaces with Color." In: *International Conference on 3D Vision (3DV)*.

Zhou, Keyang, Bharat Bhatnagar, Jan Eric Lenssen, and Gerard Pons-Moll (2022). "TOCH: Spatio-Temporal Object Correspondence to Hand for Motion Refinement." In: *European Conference on Computer Vision (ECCV)*.

Zhou, Xingyi, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei (2017). "Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach." In: *International Conference on Computer Vision (ICCV)*.

Zhou, Xingyi, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei (2016). "Deep Kinematic Pose Regression." In: *European Conference on Computer Vision (ECCV)*.

Zhou, Yi, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao (June 2019). "On the Continuity of Rotation Representations in Neural Networks." In: *Computer Vision and Pattern Recognition (CVPR)*.

Zollhöfer, Michael, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. (2014). "Real-time non-rigid reconstruction using an RGB-D camera." In: *ACM Transactions on Graphics (TOG)* 33.4.

Zou, Yuliang, Jimei Yang, Duygu Ceylan, Jianming Zhang, Federico Perazzi, and Jia-Bin Huang (2020). "Reducing Footskate in Human Motion Reconstruction with Ground Contact Constraints." In: *Winter Conference on Applications of Computer Vision (WACV)*.