Reconstruction and Synthesis of Human-Scene Interaction

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Mohamed Hassan
aus Khartoum, Sudan

Tübingen
2022

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

Tag der mündlichen Qualifikation:	10.02.2023
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter	Prof.Dr. Michael Black
2. Berichterstatter	Prof.Dr. Andreas Geiger

Abstract

Humans live within a 3D scene and constantly interact with it to perform tasks. We observe that the world constrains human movements and vice versa. We argue that the 3D scene is vital for understanding, reconstructing, and synthesizing human motion. We present several approaches which take the scene into consideration in reconstructing and synthesizing Human-Scene Interaction (HSI).

State-of-the-art pose estimation methods ignore the 3D scene and hence reconstruct poses that are inconsistent with the scene. We address this by proposing a pose estimation method that takes the 3D scene explicitly into account. We call our method PROX for Proximal Relationships with Object eXclusion. We constrain the body reconstruction by two physical scene constraints: penetration constraint, and contact constraint. We demonstrate the power of our method on a new dataset composed of 12 different 3D scenes and RGB-D sequences of 20 subjects moving in and interacting with the scenes. By running our method on the new dataset sequences; we provide pseudo ground-truth reconstruction of 3D humans interacting with 3D scenes.

We leverage the PROX data and build a method to automatically place 3D scans of people with clothing in scenes. The core novelty of our method is encoding the proximal relationships between the human and the scene in a novel HSI model, called POSA for Pose with prOximitieS and contActs. Our model is body-centric, which enables it to generalize to new scenes. POSA augments each vertex of the SMPL-X body model with (a) the contact probability with the scene surface and (b) the corresponding semantic scene label.

POSA is limited to static HSI, however. We propose a real-time method for synthesizing dynamic HSI, which we call SAMP for Scene-Aware Motion Prediction. SAMP enables virtual humans to navigate cluttered indoor scenes and naturally interact with objects. At the core of SAMP is a stochastic variational model which captures the diversity of HSI. Unlike previous deterministic methods, this allows SAMP to synthesize different high-quality motion styles of the same action. In addition, SAMP captures the different body positions and orientations in which an action could be performed on the object surface (e.g., different positions and sitting orientations on the sofa). SAMP incorporates an explicit Path Planning Module which allows collision-free navigation in cluttered environments.

Data-driven kinematic models, like SAMP, can produce high-quality motion when applied in environments similar to those shown in the dataset. However, when applied to new scenarios, kinematic models can struggle to generate realistic behaviors that respect scene constraints. In contrast, we present InterPhys which uses adversarial imitation learning and reinforcement learning to train physically-simulated characters that perform scene interaction tasks in a physical and life-like manner. These scene interactions are learned using an adversarial discriminator that evaluates the realism of a motion within the context of a scene. The key novelty involves conditioning both the discriminator and the policy networks on scene context. An efficient randomization approach of the training objects, their placements, and sizes enables our method to generalize beyond the objects and scenarios shown in the training dataset, producing natural character-scene interactions despite wide variation in object shape and placement.

Zusammenfassung

Menschen leben in einer 3D-Szene und interagieren ständig mit ihr, um Aufgaben auszuführen. Wir beobachten, dass das Umfeld menschliche Bewegungen einschränkt und umgekehrt. Daher argumentieren wir, dass die 3D-Szene für das Verständnis, die Rekonstruktion und die Synthese menschlicher Bewegungen von entscheidender Bedeutung ist. Wir stellen mehrere Ansätze vor, welche die Szene bei der Rekonstruktion und Synthese von Mensch-Szene-Interaktion (HSI) berücksichtigen.

Moderne Verfahren zur Posenschätzung ignorieren die 3D-Szene und rekonstruieren deswegen Posen, welche nicht zur Szene passen. Wir adressieren diesen Nachteil, indem wir ein Verfahren zur Posenschätzung vorschlagen, das die 3D-Szene explizit berücksichtigt. Wir nennen unsere Methode PROX für "Proximal Relationships with Object Exclusion". Wir schränken die Körperrekonstruktion durch zwei physikalische Szenenbeschränkungen ein, einerseits Penetrationsbeschränkung und als auch Kontaktbeschränkung. Wir veranschaulichen die Leistungsfähigkeit unserer Methode an einem neuen Datensatz, der aus 12 verschiedenen 3D-Szenen und RGB-D-Sequenzen von 20 Personen besteht, welche sich in die Szenen bewegen und mit ihnen interagieren. Durch Anwendung unserer Methode auf den neuen Datensatzsequenzen, bieten wir eine kuenstliche Rekonstruktion von 3D-Menschen, die mit 3D-Szenen interagieren, als Referenz.

Wir nutzen die PROX-Daten um eine Methode zu entwickeln, welche automatisch 3D-Scans von bekleideten Personen in Szenen zu platzieren. Die Neuheit unserer Methode ist die Kodierung des proximalen Zusammenhangs zwischen Mensch und Szene in einem neuen HSI-Modell names POSA, kurz fuer "Pose with prOximitieS and contActs". Unser Modell ist koerperzentriert, wodurch es auch auf neuen Szenen verallgemeinert. POSA erweitert jeden Punkt in dem SMPL-X Koerpermodell um (a) die Kontaktwahrscheinlichkeit mit der Szenenoberflaeche und (b) das entsprechende semantische Szenenlabel.

POSA ist jedoch auf statische HSI limitiert. Wir schlagen eine Echtzeitmethod zur Synthese dynamischer HSI vor, welche wir SAMP, kurz fuer "Scene-Aware Motion Prediction", nennen. SAMP ermoeglicht es virtuellen Menschen auch in ueberladenen Innenszenen zu navigieren und auch auf natuerliche Art und Weise mit Objekten zu interagieren. Im Wesentlichen ist SAMP ein stochastisches Variationsmodell, welches die Vielfalt von HSI erfasst. Anders als fruehere deterministische Methoden, kann SAMP dadurch verschiedene Bewegungsspiele derselben Aktion synthetisieren. Darueber hinaus erfasst SAMP verschiedene Koerperpositionen und -orientierungen, in denen eine Aktion auf einer Objektoberflaeche ausgefueht werden koennte, e.g., verschiedene Sitzpositionen/-orientierungen auf einem Sofa. Zusammengefasst beinhaltet SAMP ein explizites Pfadplanungsmodul, das kolisionsfreie Navigation in ueberladenen Umgebungen ermoeglicht.

Daten-getriebene kinematische Modelle, wie SAMP, koennen (realistische Bewegungen/ Bewegungen von hoher Qualitaet) generieren, wenn sie fuer Umgebungen verwendet werden, welche aehnlich zu Samples aus dem Datenset sind. Werden solche Modelle jedoch auf neuen, ungesehenen Szenarien angewendet, koennen sie Schwierigkeiten haben realistische Bewegungen zu generieren, welche Szenenbeschraenkungen respektieren. Im Gegensatz dazu stellen wir InterPhys vor,

das auf Adversarial Imitation Learning und Reinforcement Learning basiert, um physikalisch simulierte Charaktere zu trainieren, welche HSI Aufgaben auf lebensechte Weise ausfueheren. Diese Szeneinteraktionen werden mittels einem Adversarial Discriminator gelernt, welcher die realitaetsnaehe einer Bewegung innerhalb einer Szene evaluiert. Die wichtigste Neuerung besteht in der Konditionierung des Discriminators und der Policy-Netzwerken auf die entsprechend Szene. Zusaetzlich ermoeglicht eine effiziente Randomisierung der Trainings-Beispiele, in Bezug auf deren Position und Groesse, unsere Methode auch auf Objekte und Szenen ueber den Datensatz hinaus zu generalisieren. Dadurch erzeugt unsere Methode natuerliche Charakteren-Szenen-Interaktionen trotz grosser Variationen in Form und Position von Objekten.

Acknowledgement

Joining Perceiving Systems (PS) was intimidating, it was like how Michael always describes it: like drinking from a fire hose. Nonetheless, everyone in PS tried very hard to make me feel at ease and at home. Michael is a true teacher, role model, and of one the kindest people I met. I aspire that one day I have his enthusiasm and dedication to research. Submitting my first paper was nerve-racking. But Michael took a genuine interest in my academic success as well as my well-being and mental health. He kept assuring me and saying that it is his job to help me succeed. His words meant a lot to me and indeed made me feel more at ease.

I owe everything to my parents. They taught me everything I know and they made me the man I am today. The love of my siblings and the memories we share are what keep me going. My wife crossed the ocean and left everything and everyone to support my journey. She put up with my long working hours and endless deadlines. She is my safe place when the wave comes.

Michael's positive energy has definitely put its mark on the whole department. I had a lot of fun at PS and I can confidently say my time at PS was the best time of my life. Everyone seemed approachable and ready to help. The professionalism of the PS admins cannot be matched. I will always be grateful to Melanie, Nicole, and Johanna.

Dimitris was my mentor and friend. His role was pivotal in every stage of the first two projects. From brainstorming to coding to debugging to writing to presenting.

I learned a lot from my fellow Ph.D. students at PS. I am most grateful to Vassilis Choutas and Partha Ghosh. I aspire to be an ML expert like Partha and a hardcore programmer like Vassilis. Finishing my Ph.D. wouldn't have been possible if it was not for their academic and non-academic help. The moments we shared will forever live in my memory. Soubhik Sanyal and Lea Müller are the best Ph.D. neighbors and I will miss our endless coffee chats. Yassine Nemmour is a true brother, a friend in need, and part of the family. Sergey Prokudin was of a lot of help in my early days at PS and I consider him my sports guru. My time in PS was the time of my life thanks to the special moments I shared with Timo, Priyanka, Abhinanda, Ahmed, Marilyn, Anurag, Jonas, Yao, Thomas, Sergi, Silvia, Shashank, Joachim, Jinlong, Hongwei, Qianli, Muhammed, Sai, Arjun, Paul, Omri, Nikos, Viktoria, Georgios, and others.

My time at Adobe was a lot of fun and introduced me to the beautiful world of animation. Duygu Ceylan is a wonderful manager and she gave me her full support through a tough project. It was my pleasure to work with her alongside Ruben Villegas, Jun Saito, Jimei Yang, and Yi Zhou.

I am grateful to Sanja Fidler for the opportunity to intern with her great team at Nvidia. Her speed still surprises me. At Nvidia, it was a privilege to work closely with Jason Peng and I learned a lot about reinforcement learning from him. I am grateful as well to the rest of the great team Tingwu Wang and Kelly Guo.

None of this work would have been possible without the data and infrastructure built by our wonderful IT and capture team. Thanks to Benjamin, Andrea, Mason, Tsvetelina, Markos, and Tobias. My thesis has been supported by the International Max Planck Research School (IMPRS) and its amazing coordination office Leila Masri and Sara Sorce. Andreas Geiger and Goerg Martius were instrumental throughout the graduation process. I am grateful for their time, support, and feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For computers to help us, humans, and take an active role in our lives, they need to understand us and understand our movement. Thus, decades of computer vision research has been devoted to analyzing and reconstructing human motion from visual input. The research has been motivated by the wide range of applications in medicine, surveillance, autonomous systems, human-computer interaction (HCI), and other fields. Computer graphics studies the inverse problem. It is concerned with synthesizing digital visual content. Again, creating realistic-looking digital humans that move and act as humans do has always been at the heart of computer graphics

Lately, we have seen remarkable progress in capturing human motion from images. We can now segment a human from the background [79, 209], detect the 2D [24, 88, 160] or 3D [131, 151, 154] pose, and can even reconstruct the full 3D human body surface with its facial expressions and finger articulation from a single RGB image [35, 48, 94]. Likewise, reconstructing 3D scenes from images is becoming more and more tractable. Full 3D scenes can be recovered from a sequence of images [29, 40]. 3D objects and their semantics can be reconstructed from a single RGB image [68, 128, 134, 143, 206]. However, each problem is studied separately; reconstructing 3D humans and the surrounding 3D scene are rarely studied together as we present in this thesis. Similarly, the computer graphics community has built tools to create, manipulate, and animate realistic-looking human avatars and 3D scenes. However, limited research has been devoted to creating avatars that can interact with their surrounding scene. This thesis aims to bridge this gap.

Reconstructing humans from images has a broad range of applications [135]. Human Pose Estimation (HPE) is essential for all autonomous agents that are meant to interact with humans. Self-driving cars, delivery robots, and warehouse robots all need HPE to be of assistance to humans and to make sure they are not imposing risks to their lives [32, 62]. At the minimum, these agents need to know if a human is standing in front of them so as not to collide with him or her. HPE will be a key to how we interact with computers. This is most evident in games, where a player controls the game with his or her movements. Moreover, HPE is becoming widely used in controlling different appliances and even cars [18]. In healthcare, HPE is growing increasingly popular as it provides a cheap and easy-to-use mechanism to study and assess human movement [189]. For example, it is used by healthcare workers to assess the motor skills of patients and to monitor athletes' performances.

Human body models are extensively used to reason about human pose and shape. They provide a prior distribution over the human body configuration, shape, and movements. Early research in HPE represents the human pose as a set of unconnected joints or a skeleton. Although these representations are still commonly used in 2D and 3D HPE, they are very limited and fail to represent the human shape, expression, or texture. Another way to approximate the human body is to use a set of geometric primitives [42, 95]. Although these models carry more information than joints and skeleton, they remain crude approximations of the human body. In contrast, volumetric models approximate the human volume with a higher level of detail including facial expressions, finger articulation, and soft tissue deformation. Lately, we have seen several attempts toward creating volumetric models that represent the human body with clothing. Representing human hair remains a challenge. In this thesis, we make use of the SMPL-X model [153], which models the human body, hands, and face.

In 2D HPE, the pose is commonly represented as a set of keypoints representing the major joints in the human body. Pre-deep learning, different hand-crafted image features have been used to detect these keypoints. After the emergence of deep learning, researchers started tackling the problem with deep learning machinery. The problem has been formulated as a regression problem with neural networks (NN) regressing the 2D joints positions from images [195]. Neural networks have proven to be very effective for this problem, the performance of 2D HPE methods is now close to 100% in public benchmarks. 2D HPE research is not limited to estimating a single person but has expanded to cover estimating poses of multiple people [88, 160] and people under occlusion.

Unlike 2D HPE, 3D HPE is still a challenging problem with active research. Inferring 3D information from 2D input is inherently ambiguous and ill-posed. This is due to the depth ambiguity, lack of ground-truth training data, and the fact that multiple 3D poses can be projected to the same 2D pose. While ground-truth 2D poses can be annotated on images, this is not the case for the 3D poses. This is normally tackled in the literature by using additional data like multi-view images, IMUs, or markers. Inspired by success in 2D HPE, deep learning has become the de facto technique for 3D HPE. Recent methods go beyond reconstructing the major joints of the human body to reconstruct the full 3D surface of the body, hands, and face from a single RGB image [153]. The 3D surface is critical for modeling Human-Scene Interaction (HSI). It is through this surface we interact with the 3D scene. Therefore, when tackling the problem of reconstructing HSI in this thesis, we are concerned with reconstructing the full 3D surface of the body.

This thesis studies human motion and the interaction between humans and the surrounding from a computer vision perspective as well as a computer graphics perspective. Humans do not live in a vacuum; they live in, and constantly interact with, their surrounding 3D scenes. In our thesis, we argue that human motion should be studied in its context, the 3D scene. When human motion is being reconstructed, we should consider the surrounding scene as it provides vital information that facilitates and enhances reconstruction. The same argument goes for synthesis, our motion can only make sense within a 3D scene. Even a simple walking motion is only possible when

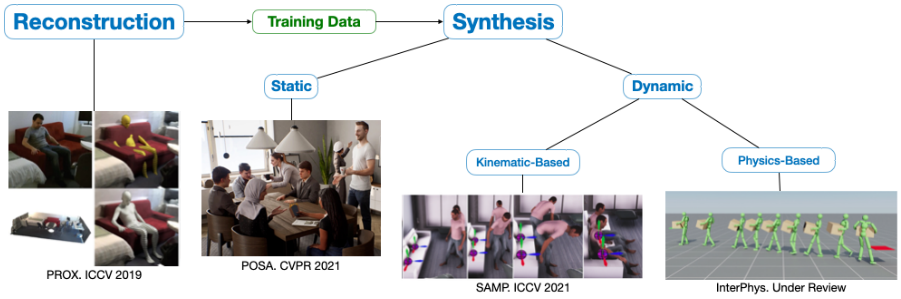


Figure 1.1: Thesis overview. See text for more detials.

a supporting plane exists. The thesis presents four connected works that build on each other to tackle the problem of reconstructing and synthesizing the interaction between a human and the surrounding 3D scene. An overview of the four works covered in this thesis is shown in Fig. 1.1.

In our first work, PROX [73], we show that current 3D human pose estimation methods produce results that are not consistent with the 3D scene. That is because they perform 3D human pose estimation without explicitly considering the scene. The key contribution of PROX is to exploit static 3D scene structure to better estimate human pose from monocular images. The method enforces *Proximal Relationships with Object eXclusion* and is called *PROX*. To test this, we collect a new dataset composed of 12 different 3D scenes and RGB sequences of 20 subjects moving in and interacting with the scenes. We represent human pose using the 3D human body model SMPL-X [153] and extend SMPLify-X [153] to estimate body pose using scene constraints. We make use of the 3D scene information by formulating two main constraints. The inter-penetration constraint penalizes intersection between the body model and the surrounding 3D scene. The contact constraint encourages specific parts of the body to be in contact with scene surfaces if they are close enough in distance and orientation. For quantitative evaluation, we capture a separate dataset with 180 RGB frames in which the ground-truth body pose is estimated using a motion capture system. We show quantitatively that introducing scene constraints significantly reduces 3D joint error and vertex error. Our code and data are available for research at https://prox.is.tue.mpg.de.

PROX provides data of 3D humans interacting with 3D scenes. We leverage this data to learn a method for synthesizing HSI. POSA, which stands for Pose with prOximitieS and contActs, is a novel Human-Scene Interaction (HSI) model that encodes proximal relationships. The goal of POSA [75] is to learn how humans interact with scenes and leverage this to enable virtual characters to do the same. The representation of interaction is body-centric, which POSA to generalize to new scenes. Specifically, POSA augments the SMPL-X parametric human body model such that, for every mesh vertex, it encodes (a) the contact probability with the scene surface and (b) the corresponding semantic scene label. We learn POSA with a VAE conditioned on the SMPL-X vertices, and train on the PROX dataset, which contains SMPL-X meshes of people interacting with 3D scenes.We demonstrate the value of POSA with two applications. First, we automatically place 3D scans of people in scenes. To do so, we use a SMPL-X model fit to the scan as a proxy and then find its most likely placement in 3D. POSA provides an effective representation to search for affordances in the scene that match the likely contact relationships for that pose. We perform a perceptual study that shows significant improvement over the state of the art on this task. Second, we show that POSA's learned representation of body-scene interaction supports monocular human pose estimation that is consistent with the 3D scene, improving on the state of the art. Our model and code are available for research purposes at https://posa.is.tue.mpg.de.

POSA is a step towards synthesizing realistic HSI. However, it is limited to synthesizing static HSI. In SAMP [74], for Scene-Aware Motion Prediction, we learn to synthesize dynamic HSI from human demonstration. Our goal is to enable virtual humans to navigate within cluttered indoor scenes and naturally interact with them. Such embodied behavior has applications in virtual reality, computer games, and robotics, while synthesized behavior can be used as training data. The problem is challenging because real human motion is diverse and adapts to the scene. For example, a person can sit or lie on a sofa in many places and with varying styles. We must model this diversity to synthesize virtual humans that realistically perform human-scene interactions. We present a novel data-driven, stochastic motion synthesis method that models different styles of performing a given action with a target object. SAMP generalizes to target objects of various geometries while enabling the character to navigate in cluttered scenes. To train SAMP, we collected MoCap data covering various sitting, lying down, walking, and running styles. We demonstrate SAMP on complex indoor scenes and achieve performance superior to existing solutions. Code and data are available for research at https://samp.is.tue.mpg.de.

SAMP is a supervised learning method that learns from human demonstration only. As a result, it struggles to generate realistic motion in scenarios that differ significantly from the training data. In addition, the training frames must be manually labeled with action labels. In contrast, the last work of this thesis, InterPhys, uses adversarial imitation learning and reinforcement learning to train physically-simulated characters that perform scene interaction tasks in a natural and life-like manner. InterPhys is able to learn natural scene interaction behaviors from large unstructured motion datasets, without manual annotation of the motion data. These scene interactions are learned using an adversarial discriminator that evaluates the realism of a motion within the context of a scene. The key novelty involves conditioning both the discriminator and the policy networks on scene context. We demonstrate the effectiveness of our approach through three challenging scene interaction tasks: carrying, sitting, and lying down, which require coordination of a character's movements in relation to objects in the environment. Our policies learn to seamlessly transition between different behaviors like idling, walking, and sitting. Using an efficient approach to randomize the training objects and their placements during training enables our method to generalize beyond the objects and scenarios in the training dataset, producing natural character-scene interactions despite wide variation in object shape and placement. The approach takes physics-based character motion generation a step closer to broad applicability.

In summary, this thesis presents four connected pieces that try to address the problem of reconstructing and synthesizing HSI. Our first work, PROX, introduces a method that uses scene constraints to improve HSI reconstruction. POSA makes use of the PROX data and learns a model to synthesize static HSI. SAMP takes HSI synthesis a step further and introduces a model for synthesizing diverse dynamic HSI. Lastly, InterPhys synthesizes physical dynamic HSI and enables generalization to new unseen scenarios. This thesis contributes a key step toward reconstructing and synthesizing the interaction between humans and their surrounding environments, and we hope it will inspire more future work in this domain.

Chapter 2

Related Work

This chapter starts by providing an overview of previous attempts in using scene constraints to improve human pose estimation. It then transitions to reviewing previous HSI synthesis work. We provide an overview of previous work that synthesizes static as well as dynamic HSI.

2.1 Reconstructing Human-Scene Interaction

The problems of reconstructing 3D humans and 3D scenes have been studied for years, albeit mostly dis-jointly. Typically, pose estimation methods [135, 136] try to recover the articulated human pose from visual inputs while ignoring the scene. Similarly, 3D scene reconstruction methods [6, 52] ignore the human while reconstructing the scene. The community has made significant progress on estimating human body pose and shape from images [20, 53, 89, 96, 124, 131, 136, 147, 152, 161, 175, 183, 231]. Recent methods, based on deep learning, extend 3D human pose estimation to complex scenarios [35, 48, 96, 131, 147, 152].

Most work represents 3D humans as skeletons [89, 183]. However, the 3D body surface is important for physical interactions. This is addressed by learned parametric 3D body models [12, 94, 123, 148, 153, 216]. In PROX [73] and POSA [75] we employ SMPL-X [153], which models the body with full face and finger articulation.

Several works focus on improving 2D object detection, 2D pose, and action recognition by observing RGB imagery of people interacting with objects [8, 64, 106, 159, 220]. Similar observations are used to reason about 3D scenes [41, 50, 65], i.e. rough 3D reconstruction and affordances [56], however scene cues are not used as feedback to improve human pose as we do in PROX [73].

Yamamoto and Yagishita [219] were the first to use scene constraints in 3D human tracking. They observed that the scene can constrain the position, velocity, and acceleration of an articulated 3D body model. Later work adds object contact constraints to the body to effectively reduce the degrees of freedom of the body and make pose estimation easier [103, 170]. Brubaker et al. [23] focus on walking and perform 3D person tracking by using a kinematic model of the torso and the lower body as a prior over human motion and conditioning its dynamics on the 2D Anthropomorphic Walker [107]. Hasler et al. [71] reconstruct a rough 3D scene from multiple unsynchronized moving cameras and employ scene constraints for pose estimation. The above methods all had the right idea but required significant manual intervention or were applied in very restricted scenarios.

Most prior methods that have used world constraints focus on interaction with a ground plane [202] or simply constrain the body to move along the ground plane [230]. Most interesting among these is the work of Vondrak et al. [202] where they exploit a game physics engine to infer human pose using gravity, motor forces, and interactions with the ground. This is a very complicated optimization, and it has not been extended beyond ground contact.

Gupta et al. [63] exploit contextual scene information in human pose estimation using a GPLVM learning framework. For an action like sitting, they take motion capture data of people sitting on objects of different heights. Then, conditioned on the object height, they estimate the pose in the image, exploiting the learned pose model.

Zanfir et al. [224] establish contact constraints between the feet and an estimated ground plane. For this, they first estimate human poses in multi-person RGB videos independently and fit a ground plane around the ankle joint positions. They then refine poses in a global optimization scheme over all frames incorporating contact and temporal constraints, as well as collision constraints, using a collision model comprised of shape primitives similar to [19, 146]. Li et al. [117] introduce a method to estimate contact positions, forces, and torques actuated by the human limbs during human-object interaction.

The 3D hand-object community has also explored similar physical constraints, such as [108, 146, 158, 166, 197, 198] to name a few. Most of these methods employ a collision model to avoid hand-object inter-penetrations with varying degrees of accuracy; using underlying shape primitives [109, 146] or decomposition in convex parts of more complicated objects [109], or using the original mesh to detect colliding triangles along with 3D distance fields [198]. Triangle intersection tests have also been used to estimate contact points and forces [166]. Most other work uses simple proximity checks [187, 197, 198] and employs an attraction term at contact points. ObMan [78] proposes an end-to-end model that exploits a contact loss and inter-penetration penalty to reconstruct hands manipulating objects in RGB images.

In summary, past work focuses either on specific body parts (hands or feet) or interaction with a limited set of objects (ground or hand-held objects). In PROX [73], for the first time, we address the full articulated body interacting with diverse, complex and full 3D scenes. Moreover, we show how using the 3D scene improves monocular 3D body pose estimation.

2.2 Synthesizing Static Human-Scene Interaction

We here review work that aims at modeling HSI and using these models to synthesize static HSI. Lin et al. [118] generate 3D skeletons sitting on 3D chairs, by manually drawing 2D skeletons and fitting 3D skeletons that satisfy collision and balance constraints. Kim et al. [100] automate this, by detecting sparse contacts on a 3D object mesh and fitting a 3D skeleton to contacts while avoiding penetrations. Kang et al. [97] reason about the physical comfort and environmental support of a 3D humanoid, through force equilibrium. Leimer et al. [113] reason about pressure, frictional forces and body torques, to generate a 3D object mesh that comfortably supports a given posed 3D body. Zheng et al. [232] map high-level ergonomic rules to low-level contact constraints and deform an object to fit a 3D human skeleton for force equilibrium. Bar-Aviv et al. [14] and Liu et al. [122] use an interacting agent to describe object shapes through detected contacts [14], or relative distance and orientation metrics [122]. Gupta et al. [65] estimate human poses "afforded" in a depicted room, by predicting a 3D scene occupancy grid, and computing support and penetration of a 3D skeleton in it. Grabner et al. [60] detect the places on a 3D scene mesh where a 3D human mesh can sit, modeling interaction likelihood with GMMs and proximity and intersection metrics. Zhu et al. [236] use FEM simulations for a 3D humanoid, to learn to estimate forces, and reason about sitting comfort.

Recent work takes a data-driven approach. Jiang et al. [92] learn to estimate human poses and object affordances from an RGB-D scene, for 3D scene label estimation. SceneGrok [176] learns action-specific classifiers to detect the likely scene places that "afford" a given action. Fisher et al. [49] use SceneGrok and interaction annotations on CAD objects, to embed 3D room scans to CAD mesh configurations. PiGraphs [177] maps pairs of {verb-object} labels to "interaction snapshots", i.e. 3D interaction layouts of objects and a human skeleton. Chen et al. [33] map RGB images to interaction snapshots, using Markov Chain Monte Carlo with simulated annealing to optimize their layout. iMapper [137] maps RGB videos to dynamic interaction snapshots, by learning "scenelets" on PiGraphs data and fitting them to videos. PHOSA [226] infers spatial arrangements of humans and objects from a single image. Cao et al. [26] map an RGB scene and 2D pose history to 3D skeletal motion, by training on video-game data. Li et al. [116] follow [208] to collect 3D human skeletons consistent with 2D/3D scenes of [186, 208], and learn to predict them from a color and/or depth image. Corona et al. [37] use a graph attention model to predict motion for objects and a human skeleton, and their evolving spatial relationships.

Another HSI variant is Hand-Object Interaction. We discuss only recent work [22, 38, 191, 203]. Brahmbhatt et al. [22] capture fixed 3D hand-object grasps, and learn to predict contact; features based on object-to-hand mesh distances outperform skeleton-based variants. For grasp generation, 2-stage networks are popular [140]. Taheri et al. [191] capture moving SMPL-X [153] humans grasping objects, and predict MANO [167] hand grasps for object meshes, whose 3D shape is encoded with BPS [162]. Corona et al. [38] generate MANO grasping given an object-only RGB image; they first predict the object shape and rough hand pose (grasp type), and then they refine the latter with contact constraints [78] and an adversarial prior.

Closer to our work, POSA [75], PSI [228] and PLACE [227] populate 3D scenes with SMPL-X [153] humans. PSI [228] trains a cVAE to estimate humans from a depth image and scene semantics. The model provides an implicit encoding of HSI. PLACE [227], on the other hand, explicitly encodes the scene shape and human-scene proximal relations with BPS [162], but does not use semantics. Unlike PSI and PLACE, POSA is a human-centric model; inherently this is more portable to new scenes. Moreover, instead of the sparse BPS distances in PLACE [227], POSA uses dense body-to-scene contact, and also exploits scene semantics similar to PSI [228].

2.3 Synthesizing Dynamic Human-Scene Interaction

Previous methods for synthesizing dynamic human-scene interaction fall under one of two categories: kinematic-based methods or physics-based methods. Kinematic-based methods synthesize motion with no regards to physics-laws. On the other hand, physics-based methods synthesize physical motion using physics simulators. This section provides an overview of both categories. In addition, we provide a brief overview of previous research in human motion synthesis in general.

2.3.1 Kinematic-Based Methods

Traditional animation methods [11, 57, 110, 199] generally edit, retarget, or replay motion clips from a database in order to synthesize motions for a given task. Motion retargetting allows retargetting of existing motions to new characters [4, 58] or new environments [101, 112]. Seminal work of Gleicher [57] proposes a technique to adapt a reference motion to a new character with the same structure but different bone lengths. The motion of the new character is recomputed such that it satisfies a set of space and time constraints, while remaining as close as possible to the original motion. The method is used to adapt character-scene-interaction motions like box carrying and climbing a ladder. While this method allows motions to be adapted to new characters, Lee and Shin [110] introduce an interactive motion editing technique that allows motions to be adapted to new characters and new environments. Lee et al. [111] arrange a large repertoire of motions in a graph. Once built, a virtual character is animated by searching the graph for the next suitable motion clip at each time step. They show that this technique can enable virtual characters to traverse a playground-like environment. Lee et al. [112] focus on traversing an environment made of a set of building blocks. Each building block is annotated with associated motion data from MoCap. The annotated building blocks are called motion patches. Rearranging these motion patches allows the generation of new interactions. Such editing and retargetting methods are limited to new scenarios that are similar to the ground-truth. Graph-based methods require a complicated procedure to build and search the graph. In addition, they are memory intensive, since a large dataset typically needs to be stored and accessed at run-time. More details about pre-deep learning real-time animation methods are in the survey by Van Welbergen et al. [200]. Lately, Holden et al. [86] propose a learned version of the motion matching algorithm by breaking it down to its basic steps and providing learning-based alternatives for each one. The resulting method does not require the storage of animation data.

Recently, deep-learning-based solutions have been proposed to the classical task of motion in-betweening [70, 99, 235] where a user provides a set of key frames and a neural network generates the in-between motion. Interpolation-based techniques create new motion by blending existing motion segments [149, 169].

Several optimization-based techniques have been proposed to compute motion from a sparse user input like motion trajectories or a set of keyframes. However, human motion is complex and high dimensional making such optimization hard and often intractable. Safonova et al. [174] address this by transforming the problem to a low-dimensional space using Principal Component Analysis (PCA) [93].

Neural networks (feed-forward networks, LSTMs, or RNNs) have been extensively applied to the motion synthesis problem [5, 51, 67, 84, 130, 192, 201]. A typical approach predicts the future motion of a character based on previous frame(s). While showing impressive results when generating short sequences, many of these methods either converge to the mean pose or diverge when tested on long sequences. A common solution is to employ scheduled sampling [15] to ensure stable predictions at test time to generate long locomotion and dancing sequences [119, 234]. We also use a similar strategy in SAMP [74] when training MotionNet, but focus on synthesizing human-scene interaction.

Only a few previous works have explored synthesizing dynamic human-scene interaction. Earlier work [111] focuses on synthesizing motions of a character in the same environment in which the motion was captured, follow-up work [7, 98, 112, 180] assembles motion sequences from a large database to synthesize interactions with new environments or characters. In a similar fashion, Agrawal et al. [7] use motion templates based on task-specific footstep plans in a given database for synthesizing new character-scene interactions. Precision [98] analyzes a given environment and a motion database to identify the different ways the environment can support the character to move. Such data-driven methods often require large databases to achieve the expressiveness of the synthesized motions at the cost of expensive nearest neighbor matching. Hence, they do not scale well with the size and complexity of the training data.

An important sub-category of human-scene interaction involves locomotion, where the character must respond to changes in terrain with appropriate foot placement. Phase-functioned neural networks [85] have shown impressive results by using a guiding signal representing the state of the motion cycle (i.e., phase). Introducing the phase increased the expressiveness of the model and led to the generation of high-quality motion in real-time. Zhang et al. [225] extend this idea to use a mixture of experts [46, 90, 223] as the motion prediction network. An additional gating network is used to predict the experts' blending weights at run time.

More closely related to our work is the Neural State Machine (NSM) [188], which extends the ideas of phase labels and expert networks to model HSI such as sit, carry, and open. While NSM is a powerful method, it does not generate variations in such interactions, which is one of the contributions of our work SAMP [74]. Our experiments also demonstrate that NSM often fails to avoid intersections between the 3D character and objects in cluttered scenes. Furthermore, training NSM requires time-consuming manual, and often ambiguous, labeling of the phase. We find that using scheduled sampling [15] provides an alternative to generate smooth transitions without phase labels.

More recently, Wang et al. [205] introduce a hierarchical framework for synthesizing HSI. They generate sub-goal positions in the scene, predict the pose at each of these sub-goals, and synthesize the motion between such poses. This method requires a post-optimization framework to ensure smoothness and robust foot contact and to discourage penetration with the scene. Corona et al. [36] use a semantic graph to model human-object relationships, followed by an RNN to predict human and object movements.

Several works have focused on modeling the stochastic nature of human motion, with a specific emphasis on trajectory prediction. Both in terms of multi-modal trajectory modeling as well as pose diversity. Given the past trajectory of a character, they model multiple plausible future trajectories [10, 25, 28, 66, 126, 173, 181]. See [171] for a survey. Other work models the social aspect of human behaviors by predicting socially plausible trajectories [10, 66]. Only a few works takes into consideration the scene context to predict physically and socially plausible trajectories [173]. Recently, Cao et al. [25] sample multiple future goals and then use them to generate different future skeletal motions. This is similar in spirit to our use of GoalNet in SAMP. The difference is that our goal is to predict various trajectories that always lead to the same target object (instead of predicting any plausible future trajectory). In addition, we use the predicted trajectory to guide the process of generating full human body motion while these works predict the trajectory only.

Modeling the stochasticity of the full human motion is a less explored area [210, 221, 222]. DLow [221] improves diversity of the predicted skeletal motion by an explicit diversity-promoting prior. Yuan and Kitani propose a diversity sampling function [222]. Wang et al. [210] model the distribution of the character's next state given its past state and the hidden variables of an RNN. Similarly, Motion VAE [119] predicts a distribution of the next poses instead of one pose using the latent space of a conditional variational auto-encoder. A separate RL policy is trained to control the latent space, hence controlling the character motion to achieve certain tasks. Instead of using a separate RL policy to control the character, MoGlow is a controllable probabilistic generative model based on normalizing flows [81]. Generating diverse dance motions from music has also been recently explored [114, 115]. Xu et al. [217] generate diverse motions by blending short sequences from a database. All the previous work focus on generating diverse locomotion, dancing, or gymnastics. To the best of our knowledge, no previous work has tackled the problem of generating diverse HSI as we do in SAMP [74].

2.3.2 Physics-Based Methods

Early physics-based methods generate motions by numerically integrating equations of motion derived from the physical models of the character [163]. The physical plausibility of the generated motion is guaranteed, but the resulting behaviors may not be particularly life-like. Heuristics, such as symmetry, stability, and power minimization [163, 204] can be incorporated into controllers to improve the realism of simulated motions.

Imitation learning is another popular approach to improve the realism of physically simulated characters. In this approach, a character learns to perform various behaviors by imitating reference motion data. Motion tracking is one of the most commonly used techniques for motion imitation and is effective at reproducing a large array of challenging skills [16, 34, 207, 212]. At its core; motion tracking uses a tracking objective that encourages the simulated character to follow a particular reference motion clip [155]. However, it can be difficult to apply tracking-based methods to solve tasks that require composition of diverse behaviors, since the tracking-objective is typically only applied with respect to one reference motion at a time.

Inspired by Generative Adversarial Imitation Learning (GAIL) [83], Peng et al. [157] train a motion discriminator on large unstructured datasets and use it as a general motion prior for training a control policy. This technique allows characters to imitate and compose behaviors from large motion datasets, without requiring any annotation of the motion clips, such as skill or phase labels. A similar GAN-like approach was concurrently introduced by Xu and Karamouzas [218]. In InterPhys [77], we leverage an adversarial imitation learning approach similar to Peng et al. [157], but go beyond prior work to develop control policies for character-scene interaction tasks. For more details on physics-based and kinematic-based animation methods, see the survey by Mourot et al. [139].

Very little work has tackled the problem of synthesizing physical character-scene interactions. Early work simplifies the object manipulation problem by explicitly attaching an object to the hands of the character [39, 138, 156], thereby removing the need for the character to grasp and manipulate an object's movements via contact. Liu and Hodgins [120] use a framework based on trajectory optimization to learn basketball dribbling. They decouple the controller into an arm controller and a locomotion controller. Chao et al. [31] propose a hierarchical controller to synthesize sitting motions. This is accomplished by dividing the sitting task into sub-tasks and training separate controllers to imitate relevant reference motion clips for each sub-task. The sub-tasks are walk, turn right, turn lift, and sit. A meta controller is then trained to select which sub-task to execute at each time step. They report a success rate of $\sim 17\%$. Similar hierarchical approaches are used to train characters to play a simplified version of football [87, 121], and to simulate fencing and boxing [213]. Merel et al. [132] train a collection of policies, each of which imitates a motion clip depicting a box-carrying or ball-catching task. The different controllers are then distilled into a single latent variable model that can then be used to construct a hierarchical controller for performing more general instances of the tasks. Concurrently, Eom et al. [47] propose a model predictive controller to solve a similar task. In contrast to the prior work, InterPhys [77] is not hierarchical, generalizes to more objects and scenes, can be trained on large datasets without manual labels, and is easily applicable to multiple tasks.

Chapter 3

Resolving 3D Human Pose Ambiguities with 3D Scene Constraints

Humans move through, and interact with, the 3D world. The world limits this movement and provides opportunities (affordances) [55]. In fact, it is through contact between our feet and the environment that we are able to move at all. Whether simply standing, sitting, lying down, walking, or manipulating objects, our posture, movement, and behavior is affected by the world around us. Despite this, most work on 3D human pose estimation from images ignores the world and our interactions with it.

Here we formulate human pose estimation differently, making the 3D world a first class player in the solution. Specifically, we estimate 3D human pose from a *single RGB image* conditioned on the 3D scene. We show that the world provides constraints that make the 3D pose estimation problem easier and the results more accurate.

We follow two key principles to estimate 3D pose in the context of a 3D scene. First, from intuitive physics, two objects in 3D space cannot *inter-penetrate* and share the same space. Thus, we penalize poses in which the body inter-penetrates scene objects. We formulate this "exclusion principle" as a differentiable loss function that we incorporate into the SMPLify-X pose estimation method [153].

Second, physical interaction requires *contact* in 3D space to apply forces. To exploit this, we use the simple heuristic that certain areas of the body surface are the most likely to contact the scene, and that, when such body surfaces are close to scene surfaces, and have the same orientation, they are likely to be in contact. Although these ideas have been explored to some extent by the 3D hand-object estimation community [109, 146, 158, 166, 197, 198] they have received less attention in work on 3D body pose. We formulate a term that implements this contact heuristic and find that it improves pose estimation.

Our method extends SMPLify-X [153], which fits a 3D body model "top down" to "bottom up" features (e.g. 2D joint detections). We choose this optimization-based framework over a direct regression method (deep neural network) because it is more

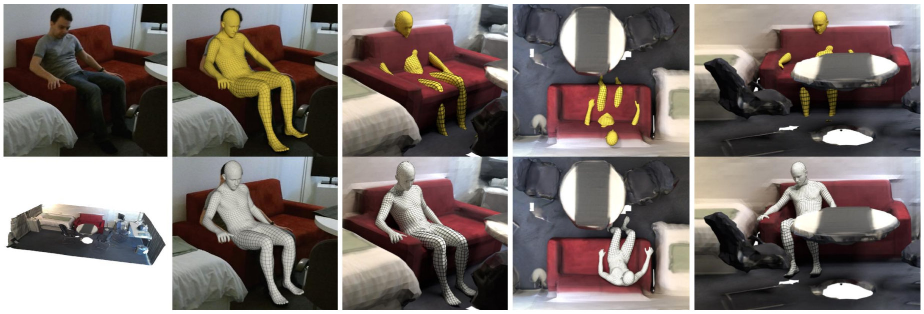


Figure 3.1: Standard 3D body estimation methods predict bodies that may be inconsistent with the 3D scene, even though the results may look reasonable from the camera viewpoint. To address this, we exploit the 3D scene structure and introduce *scene constraints* for *contact* and *inter-penetration*. From left to right: (1) RGB image (top) and 3D scene reconstruction (bottom), (2) overlay of estimated bodies on the original RGB image without (yellow) and with (gray) scene constraints, 3D rendering of both the body and the scene from (3) the camera view, (4) a top view and (5) a side view.

straightforward to incorporate our physically-motivated constraints. The method enforces *Proximal Relationships with Object eXclusion* and is called *PROX*. Figure 3.1 shows a representative example where the human body pose is estimated with and without our scene constraints. From the viewpoint of the camera, both solutions look good and match the 2D image but, when placed in a scan of the 3D scene, the results without scene constraints can be grossly inaccurate. Adding our constraints to the optimization reduces inter-penetration and encourages appropriate contact.

One may ask why such constraints are not typically used? One key reason is that to estimate and reason about contact and inter-penetration, one needs both a model of the 3D scene and a *realistic* model of the human body. The former is easy to obtain today with many scanning technologies but, if the body model is not accurate, it does not make sense to reason about contact and inter-penetration. Consequently, we use the SMPL-X body model [153], which is realistic enough to serve as a “proxy” for the real human in the 3D scene. In particular, the feet, hands, and body of the model have realistic shape and degrees of freedom.

Here, we assume that a rough 3D model of the scene is available. It is fair to ask whether it is realistic to perform monocular human pose estimation but assume a 3D scene? We argue that it is for two key reasons. First, scanning a scene today is quite easy with commodity sensors. If the scene is static, then it can be scanned once, enabling accurate body pose estimation from a single RGB camera; this may be useful for surveillance, industrial, or special-effects applications. Second, methods to estimate 3D scene structure from a single image are advancing rapidly. There are now good methods to infer 3D depth maps from a single image [45], as well as methods that do more semantic analysis and estimate 3D CAD models of the objects in the scene [142]. Our work is complementary to this direction, and we believe that monocular 3D scene estimation and monocular 3D human pose estimation should happen together.

The work here provides a clear example of why this is valuable.

To evaluate PROX, we use three datasets: two *qualitative datasets* and a *quantitative dataset*. The qualitative datasets contain: 3D scene scans, monocular RGB-D videos and pseudo ground-truth human bodies. The pseudo ground-truth is extracted from RGB-D by extending SMPLify-X to use both RGB and depth data to fit SMPL-X.

In order to get true ground-truth for the quantitative dataset, we set up a living room in a marker-based motion capture environment, scan the scene, and collect RGB-D images in addition to the MoCap data. We fit the SMPL-X model to the MoCap marker data using MoSh++ [124] and this provides ground-truth 3D body shape and pose. This allows us to quantitatively evaluate our method.

Our datasets and code are available for research at `https://prox.is.tue.mpg.de`.

3.1 Method

3.1.1 3D Scene Representation

To study how people interact with a scene, we first need to acquire knowledge about it, i.e. to perform scene reconstruction. Since physical interaction takes place through surfaces, we chose to represent the scene as a 3D mesh $M_s = (V_s, F_s)$, with $|V_s| = N_s$ vertices $V_s \in \mathbb{R}^{(N_s \times 3)}$ and triangular faces F_s. We assume a static 3D scene and reconstruct M_s with a standard commercial solution; the Structure Sensor [145] camera and the Skanect [3] software. We chose the scene frame to represent the world coordinate frame; both the camera and the human model are expressed w.r.t. this coordinate frame as explained in Sections 3.1.2 and 3.1.3, respectively.

3.1.2 Camera Representation

We use a Kinect-One camera [1] to acquire RGB and depth images of a person moving and interacting with the scene. We use a publicly available tool [2] to estimate the intrinsic camera parameters K_c and to capture synchronized RGB-D images. For each time frame t, we capture a 512×424 depth image Z^t and 1920×1080 RGB image I^t at 30 FPS. We then transform the RGB-D data into point cloud P^t.

To perform human MoCap w.r.t. to the scene, we first need to register the RGB-D camera to the 3D scene. We assume a static camera and estimate the extrinsic camera parameters, i.e. the camera-to-world rigid transformation $T_c = (R_c, t_c)$, where $R_c \in SO(3)$ is a rotation matrix and $t_c \in \mathbb{R}^3$ is a translation vector. For each sequence, a human annotator annotates 3 correspondences between the 3D scene M_s and the point cloud P^t to get an initial estimate of T_c, which is then refined using ICP [17, 233]. The camera extrinsic parameters (R_c, t_c) are fixed during each recording (Section 3.1.4),

The human body b is estimated in the camera frame and needs to be registered to the scene by applying T_c to it too. For simplicity of notation, we use the same symbols for the camera c and body b after transformation to the world coordinate frame.

3.1.3 Human Body Model

We represent the human body using SMPL-X [153]. SMPL-X is a generative model that captures how the human body shape varies across a human population, learned from a corpus of registered 3D body, face and hand scans of people of different sizes, genders and nationalities in various poses. It goes beyond similar models [12, 72, 123, 168] by holistically modeling the body with facial expressions and finger articulation, which is important for interactions.

SMPL-X is a differentiable function $M_b(\beta, \theta, \psi, \gamma)$ parameterized by shape β, pose θ, facial expressions ψ and translation γ. Its output is a 3D mesh $M_b = (V_b, F_b)$ for the human body, with $N_b = 10475$ vertices $V_b \in \mathbb{R}^{(N_b \times 3)}$ and triangular faces F_b. The shape parameters $\beta \in \mathbb{R}^{10}$ are coefficients in a low-dimensional shape space learned from approximately 4000 registered CAESAR [165] scans. The pose of the body is defined by linear blend skinning with an underlying rigged skeleton, whose 3D joints $J(\beta)$ are regressed from the mesh vertices. The skeleton has 55 joints in total; 22 for the main body (including a global pelvis joint), 3 for the neck and the two eyes, and 15 joints per hand for finger articulation. The pose parameters $\theta = (\theta_b, \theta_f, \theta_h)$ are comprised of $\theta_b \in \mathbb{R}^{66}$ and $\theta_f \in \mathbb{R}^9$ parameters in axis-angle representation for the main body and face joints respectively, with 3 degrees of freedom (DOF) per joint, as well as $\theta_h \in \mathbb{R}^{12}$ pose parameters in a lower-dimensional pose space for finger articulation of both hands, captured by approximately 1500 registered hand scans [168]. The pose parameters θ and translation vector $\gamma \in \mathbb{R}^3$ define a function that transforms the joints a long the kinematic tree $R_{\theta\gamma}$. Following the notation of [19] we denote posed joints with $R_{\theta\gamma}(J(\beta)_i)$ for each joint i.

3.1.4 Human MoCap from Monocular Images

To fit SMPL-X to a single RGB image, we employ SMPLify-X [153] and extend it to include human-world interaction constraints to encourage contact and discourage inter-penetrations. We name our method *PROX* for *Proximal Relationships with Object eXclusion*. In addition, we extend SMPLify-X to SMPLify-D, which uses both RGB and an additional depth input for more accurate registration of human poses to the 3D scene. We also extend PROX to use RGB-D input instead of RGB only; we call this configuration PROX-D.

Inspired by [153], we formulate fitting SMPL-X to monocular images as an optimization problem, where we seek to minimize the objective function

$$\begin{aligned} E(\beta, \theta, \psi, \gamma, M_s) = & E_J + \lambda_D E_D + \lambda_{\theta_b} E_{\theta_b} + \lambda_{\theta_f} E_{\theta_f} + \\ & \lambda_{\theta_h} E_{\theta_h} + \lambda_\alpha E_\alpha + \lambda_\beta E_\beta + \lambda_\mathcal{E} E_\mathcal{E} + \\ & \lambda_\mathcal{P} E_\mathcal{P} + \lambda_\mathcal{C} E_\mathcal{C} \end{aligned} \tag{3.1}$$

where θ_b, θ_f and θ_h are the pose vectors for the body, face (neck, jaw) and the two hands respectively, $\theta = \{\theta_b, \theta_f, \theta_h\}$ is the full set of optimizable pose parameters, γ denotes the body translation, β the body shape and ψ the facial expressions, as described in Section 3.1.3. $E_J(\beta, \theta, \gamma, K, J_{est})$ and $E_D(\beta, \theta, \gamma, K, Z)$ are data terms that are described below; E_J is the RGB data term used in all configurations, while E_D is the optional depth data term which is used whenever depth data is available. The terms $E_{\theta_h}(\theta_h)$, $E_{\theta_f}(\theta_f)$, $E_{\mathcal{E}}(\mathcal{E})$ and $E_\beta(\beta)$ are $L2$ priors for the hand pose, facial pose, facial expressions and body shape, penalizing deviation from the neutral state. Following [19, 153] the term $E_\alpha(\theta_b) = \sum_{i \in (elbows, knees)} \exp(\theta_i)$ is a prior penalizing extreme bending only for elbows and knees, while $E_{\theta_b}(\theta_b)$ is a VAE-based body pose prior called VPoser introduced in [153]. The term $E_{\mathcal{C}}(\beta, \theta, \gamma, M_s)$ encourages contact between the body and the scene as described in Section 3.1.5. The term $E_{\mathcal{P}}(\theta, \beta, M_s)$ is a penetration penalty modified from [153] to reason about both self-penetrations and human-scene inter-penetrations, as described in Section 3.1.6. The terms E_J, E_{θ_b}, E_{θ_h}, E_α, E_β and weights λ_i are as described in [153]. The weights λ_i denote steering weights for each term. They were set empirically in an annealing scheme similar to [153].

For the *RGB data term* E_J we use a re-projection loss to minimize the weighted robust distance between 2D joints $J_{est}(I)$ estimated from the RGB image I and the 2D projection of the corresponding posed 3D joints $R_{\theta\gamma}(J(\beta)_i)$ of SMPL-X, as defined for each joint i in Section 3.1.3. Following the notation of [19, 153], the data term is

$$E_J(\beta, \theta, \gamma, K, J_{est}) = \sum_{joint\ i} \kappa_i \omega_i \rho_J(\mathit{\Pi}_K(R_{\theta\gamma}(J(\beta)_i) - J_{est,i}) \tag{3.2}$$

where $\mathit{\Pi}_K$ denotes the 3D to 2D projection with intrinsic camera parameters K. For the 2D detections we rely on OpenPose [24, 184, 211], which provides body, face and hands keypoints jointly for each person in an image. To account for noise in the detections, the contribution of each joint in the data term is weighted by the detection confidence score ω_i, while κ_i are per-joint weights for annealed optimization, as described in [153]. Furthermore, ρ_J denotes a robust Geman-McClure error function [54] for down-weighting noisy detections.

The *depth data term* E_D minimizes the discrepancy between the visible body vertices $V_b^v \subset V_b$ and a segmented point cloud P^t that belongs only to the body and not the static scene. For this, we use the body segmentation mask from the Kinect-One SDK. Then, E_D is defined as

$$E_D(\beta, \theta, \gamma, K, Z) = \sum_{p \in P^t} \rho_D(\min_{v \in V_b^v} \|v - p\|) \tag{3.3}$$

where ρ_D denotes a robust Geman-McClure error function [54] for down weighting vertices V_b^v that are far from P^t.

3.1.5 Contact Term

Using the RGB term E_J without reasoning about human-world interaction might result in physically implausible poses, as shown in Figure 3.1; However, when humans interact with the scene they come in *contact* with it, e.g. feet contact the floor while standing or walking. We therefore introduce the term $E_{\mathcal{C}}$ to encourage contact and *proximity* between body parts and the scene around contact areas.

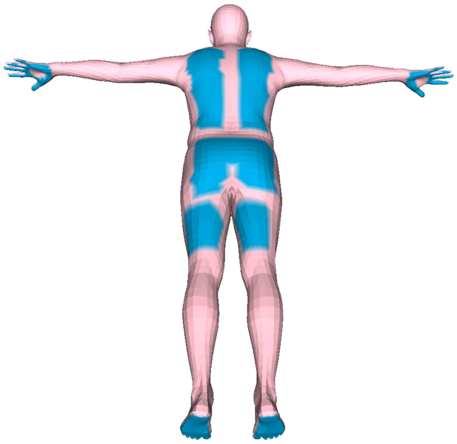


Figure 3.2: Annotated vertices that come frequently in contact with the world, highlighted with blue color.

To that end, we annotate a set of candidate contact vertices $V_{\mathcal{C}} \subset V_b$ across the whole body that come frequently in contact with the world, focusing on the actions of walking, sitting and touching with hands. We annotate 1121 vertices across the whole body, as shown in Figure 3.2. We define the contact vertices as: 725 vertices for the hands, 62 vertices for the thighs, 113 for the gluteus, 222 for the back, and 194 for the feet. $E_{\mathcal{C}}$ is defined as:

$$E_{\mathcal{C}}(\beta, \theta, \gamma, M_s) = \sum_{v_{\mathcal{C}} \in V_{\mathcal{C}}} \rho_{\mathcal{C}}(\min_{v_s \in V_s} \|v_{\mathcal{C}} - v_s\|) \tag{3.4}$$

where $\rho_{\mathcal{C}}$ denotes a robust Geman-McClure error function [54] for down-weighting vertices in $V_{\mathcal{C}}$ that are far from the nearest vertices in V_s of the 3D scene M_s.

3.1.6 Penetration Term

Intuitive physics suggests that two objects can not share the same 3D space. However, human pose estimation methods might result in self-penetrations or bodies penetrating surrounding 3D objects, as shown in Figure 3.1. We therefore introduce a penetration term that combines $E_{\mathcal{P}_{self}}$ and $E_{\mathcal{P}_{inter}}$ that are defined below:

$$\begin{aligned} E_{\mathcal{P}}(\theta, \beta, \gamma, M_s) = \\ E_{\mathcal{P}_{self}}(\theta, \beta) + E_{\mathcal{P}_{inter}}(\theta, \beta, \gamma, M_s) \end{aligned} \tag{3.5}$$

For *self-penetrations* we follow the approach of [13, 153, 198], that follows local reasoning. We first detect a list of colliding body triangles $\mathcal{P}_{self}$ using Bounding Volume Hierarchies (BVH) [194] and compute local conic 3D distance fields Ψ. Penetrations are then penalized according to the depth in Ψ. For the exact definition of Ψ and $E_{\mathcal{P}_{self}}(\theta, \beta)$ we refer the reader to [13, 198].

For body-scene *inter-penetrations* local reasoning at colliding triangles is not enough, as the body might be initialized deep inside 3D objects or even outside the 3D scene. To resolve this, we penalize all penetrating vertices using the signed distance field (SDF) of the scene M_s. The distance field is represented with a uniform voxel grid with size $256 \times 256 \times 256$, that spans a padded bounding box of the scene. Each voxel cell c_i stores the distance from its center $p_i \in \mathbb{R}^3$ to the nearest surface point $p_i^s \in \mathbb{R}^3$ of M_s with normal $n_i^s \in \mathbb{R}^3$, while the sign is defined according to the relative orientation of the vector $p_i - p_i^s$ w.r.t. n_i^s as

$$sign\left(c_i\right) = sign\left(\left(p_i - p_i^s\right) \cdot n_i^s\right); \tag{3.6}$$

a positive sign means that the body vertex is outside the nearest scene object, while a negative sign means that it is inside the nearest scene object and denotes penetration. In practice, during optimization, we can find how each body vertex V_{b_i} is positioned relative to the scene by reading the signed distance $d_i \in \mathbb{R}$ of the voxel it falls into. Since the limited grid resolution influences discretization of the 3D distance field, we perform trilinear interpolation using the neighboring voxels similar to [91]. Then we resolve body-scene inter-penetration by minimizing the loss term

$$E_{\mathcal{P}_{inter}} = \sum_{d_i < 0} \|d_i n_i^s\|^2. \tag{3.7}$$

3.1.7 Optimization

We optimize Equation 3.1 similar to [153]. More specifically, we implement our model in PyTorch and use the Limited-memory BFGS optimizer (L-BFGS) [144] with strong Wolfe line search.

3.2 Datasets

3.2.1 Qualitative Datasets

The qualitative datasets, PiGraphs and PROX, contain: 3D scene scans and monocular videos of people interacting with the 3D scenes. They do not include ground-truth bodies; thus we cannot evaluate our method quantitatively on these datasets.

PiGraphs dataset

This dataset was released as part of the work of Sava *et al.* [177]. The dataset has several 3D scene scans and RGB-D videos. It suffers from multiple limitations; the color and depth frames are neither synchronized nor spatially calibrated, making it hard to use both RGB and depth. The human poses are rather noisy and are not well registered into the 3D scenes, which are inaccurately reconstructed. The dataset has a low frame rate of 5 fps, it is limited to only 5 subjects and does not have ground-truth human motion.

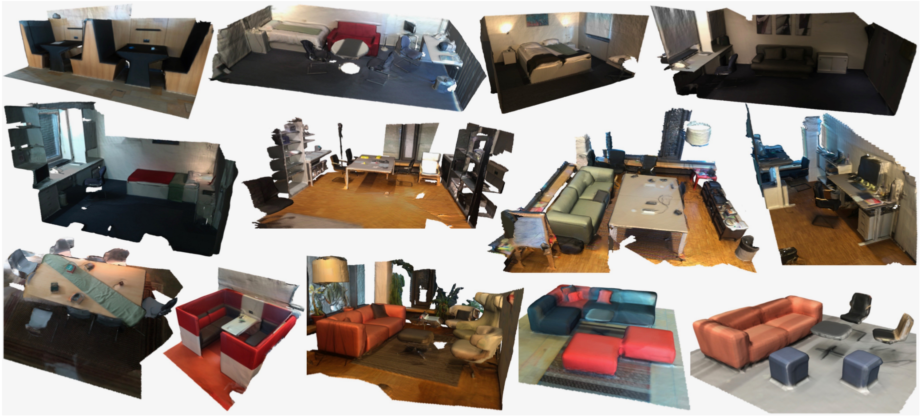


Figure 3.3: Reconstructed 3D scans of the 12 indoor scenes of our PROX dataset, as well as an additional scene from our quantitative dataset, shown at the bottom right corner.

PROX dataset

We collected this dataset to overcome the limitations of the PiGraphs dataset. We employ the commercial Structure Sensor [145] RGB-D camera and the accompanying 3D reconstruction solution Skanect [3] and reconstruct 12 indoor scenes, shown in Figure 3.3. The scenes can be grouped to: 3 bedrooms, 5 living rooms, 2 sitting booths and 2 offices. We then employ a Kinect-One [1] RGB-D camera to capture 20 subjects (4 females and 16 males) interacting with these scenes. Subjects gave written informed consent to make their data available for research purposes. The dataset provides 100K synchronized and spatially calibrated RGB-D frames at 30 fps. Figure 3.4 shows example RGB frames from our dataset. We leverage the RGB-D videos to get pseudo ground-truth of 3D humans interacting with their surrounding environment. More specifically, we fit SMPL-X meshes to the RGB-D videos while considering the scene constraints as shown in Equation 3.1.

3.2.2 Quantitative Dataset

Neither our PROX dataset nor PiGraphs [177] have ground-truth for quantitative evaluation. To account for this, we captured a separate *quantitative dataset* with 180 static RGB-D frames in sync with a 54 camera Vicon system. We placed markers on the body and the fingers. We placed everyday furniture and objects inside the Vicon area to mimic a living room, and performed 3D reconstruction of the scene, shown in the bottom right corner of Figure 3.3 with the Structure Sensor [145] and Skanect [3] similar to above. We then use MoSh++ [124] which is a method that converts MoCap data into realistic 3D human meshes represented by a rigged body model. Example RGB frames are shown in Figure 3.5 (left), while our mesh pseudo ground-truth is shown with aqua blue color. Our datasets are available for research purposes.

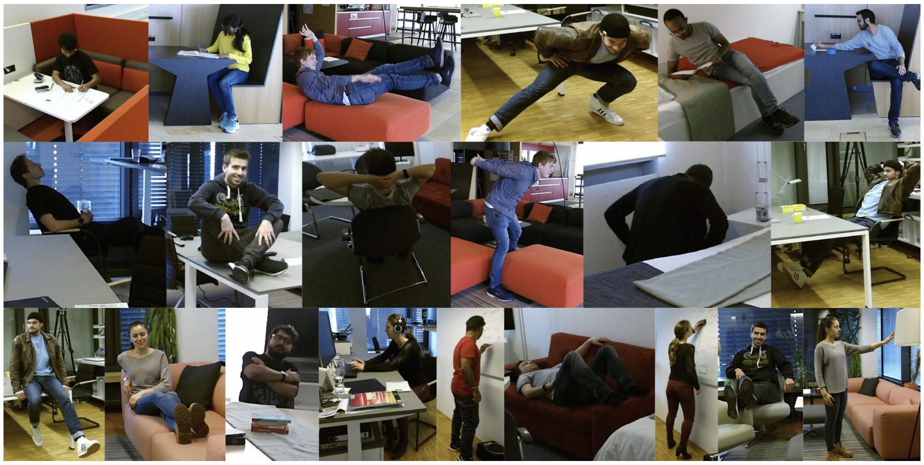


Figure 3.4: Example RGB frames of our PROX dataset showing people moving in natural indoor scenes and interacting with them. We reconstruct in total 12 scenes and capture 20 subjects. Figure 3.3 shows the 3D reconstructions of our indoor scenes.

3.3 Experiments

3.3.1 Quantitative Evaluation

To evaluate the performance of our method, as well as to evaluate the importance of different terms in Equation 3.1, we perform quantitative evaluation in Table 3.1. As performance metrics, we report the mean per-joint error without and with Procrustes alignment noted as "PJE" and "p.PJE" respectively, as well as the mean vertex-to-vertex error noted similarly as "V2V" and "p.V2V". Each row in the table shows a setup that includes different terms as indicated by the check-boxes. Table 3.1 includes two sub-tables for different datasets. **Table 1 (a):** We employ our new *quantitative dataset* with mesh pseudo ground-truth based on Vicon and MoSh++ [124], as described in Section 3.2. The first row with only E_J is an RGB-only baseline similar to SMPLify-X [153], that we adapt to our needs by using a fixed camera and estimating body translation γ, and gives the biggest "PJE" and "V2V" error. In the second row we add only the contact term $E_{\mathcal{C}}$, while in the third row we add only the penetration term $E_{\mathcal{P}}$. In both cases, the error drops a bit, however the drop is significantly bigger for the fourth row that includes both $E_{\mathcal{C}}$ and $E_{\mathcal{P}}$; this corresponds to *PROX* and achieves 167.08 mm "PJE" and 166.51 mm "V2V" error. This suggests that both $E_{\mathcal{C}}$ and $E_{\mathcal{P}}$ contribute to accuracy and are complementary. To inform the upper bound of performance, in the fifth row we employ an RGB-D baseline with E_J and E_D, which corresponds to SMPLify-D as described in Section 3.1.4. All terms of Equation 3.1 are employed in the last row; we call this configuration PROX-D. We observe that using scene constraints boosts the performance even when the depth is available. This gives the best overall performance, but PROX (fourth row) achieves reasonably good performance with less input data, i.e. using RGB only. **Table 1 (b):** We chose 4

		Eq. 3.1 terms				Erorr				
		E_J	$E_{\mathcal{C}}$	$E_{\mathcal{P}}$	E_D	PJE	V2V	PJE	V2V	
(a)	RGB	✓	✗	✗	✗	220.27	218.06	73.24	60.80	mm
	RGB + Contact	✓	✓	✗	✗	208.03	208.57	72.76	60.95	
	RGB + Penetration	✓	✗	✓	✗	190.07	190.38	73.73	62.38	
	PROX	✓	✓	✓	✗	167.08	166.51	71.97	61.14	
	SMPLify-D	✓	✗	✗	✓	72.91	69.89	55.53	48.86	
	PROX-D	✓	✓	✓	✓	**68.48**	**60.83**	**52.78**	**47.11**	
(b)	RGB	✓	✗	✗	✗	232.29	227.49	66.02	53.15	mm
	PROX	✓	✓	✓	✗	**144.60**	**156.90**	**65.04**	**52.60**	

Table 3.1: Ablation study for Equation 3.1; each row contains the terms indicated by the check-boxes. Units in mm. **Table (a):** Evaluation on our *quantitative dataset* using mesh pseudo ground-truth based on Vicon and MoSh++ [124]. **Table (b):** Evaluation on chosen sequences of our *qualitative dataset* using pseudo ground-truth based on SMPLify-D. **Tables (a, b):** We report the mean per-joint error without/with procrustes alignment noted as "PJE" / "p.PJE", and the mean vertex-to-vertex error noted as "V2V" / "p.V2V".

random sequences of our new PROX dataset. We generate pseudo ground-truth with PROX-D, which uses both RGB and depth. We show a comparison between the RGB-only baseline (first row) and PROX (second row) compared to the pseudo ground-truth of PROX-D. The results support the above finding that the scene constraints in PROX contribute significantly to accuracy.

3.3.2 Qualitative Evaluation

In Figure 3.5 we show qualitative results for our *quantitative dataset*. Furthermore, we show representative qualitative results on our qualitative dataset in Figure 3.6. Qualitative results on PiGraphs dataset are shown in Figure 3.7. In both figures, the lack of scene constraints (yellow) results in severe penetrations in the scene. Our method, PROX, includes scene constraints (light gray) and estimates bodies that are significantly more consistent with the 3D scene, i.e. with realistic contact and without penetrations.

Figures 3.8-3.10 show additional qualitative results for our method (light gray) on our PROX dataset and compare it to the RGB-only baseline (yellow). For each example, we show from left to right: (1) RGB image, (2) renderings from different viewpoints.

Figure 3.11 shows additional qualitative results for our method (light gray) on the *PiGraphs* dataset [177] and compares it to the RGB-only baseline (yellow). Please note that *PiGraphs* [177] estimates just a 3D skeleton of only the major body joints. In contrast, we estimate a full 3D mesh, and include facial expressions and finger articulation. The mesh representation of our realistic human model helps to better reason about proximity to the world, contact and penetrations. For each example, we show from left to right: (1) RGB image (2) renderings from different viewpoints.

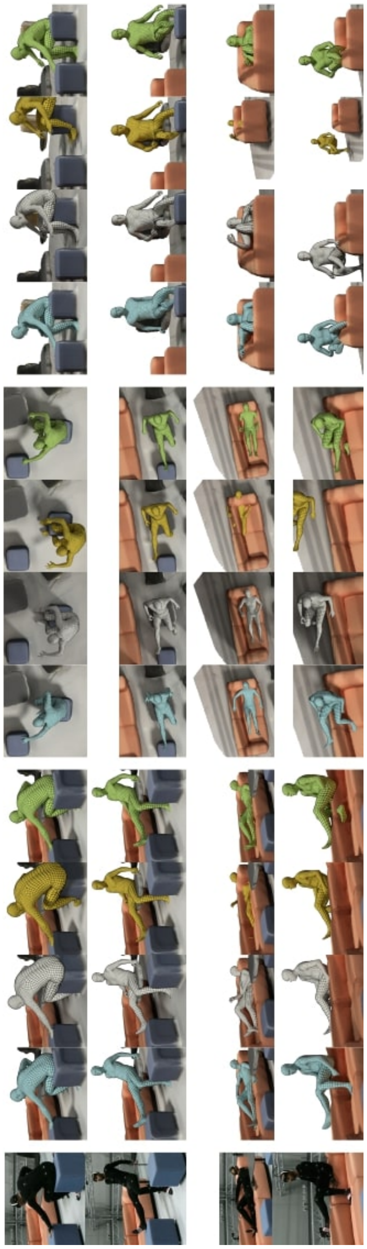


Figure 3.5: Examples from our *quantitative dataset*, described in Section 3.3. From left to right: (1) RGB images, (2) rendering of the fitted model and the 3D scene from the camera viewpoint; aqua blue for the mesh pseudo ground-truth, light gray for the results of our method PROX, yellow for results without scene constraints, green for SMPLify-D, (3) top view and (4) side view.

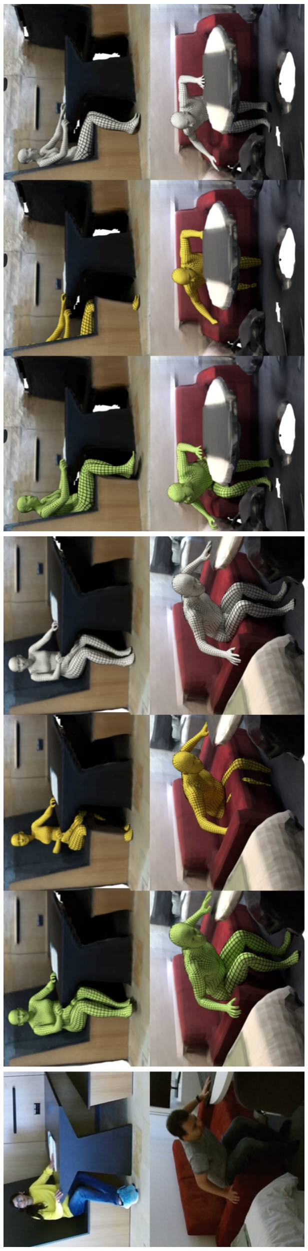


Figure 3.6: Qualitative results of our method on our *qualitative dataset*. From left to right: (1) RGB images, (2) rendering from the camera viewpoint; light gray for the results of our method PROX, yellow for results without scene constraints, and green for SMPLify-D, (3) rendering from a different view, that shows that the camera view is deceiving.

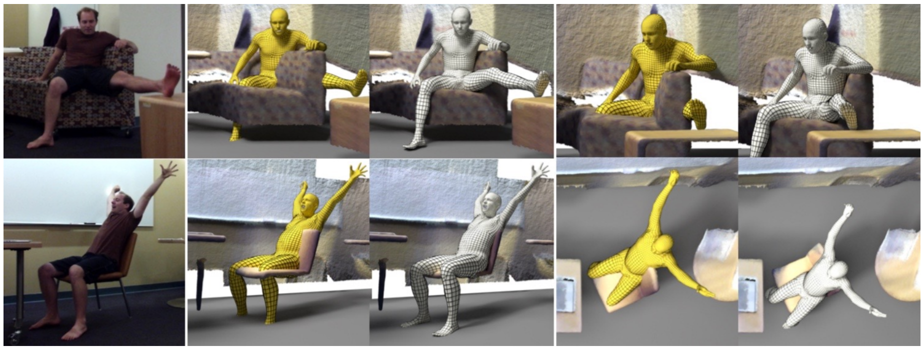


Figure 3.7: Qualitative results of our method on the PiGraphs dataset [177]. From left to right: (1) RGB images, (2) rendering from the camera viewpoint; light gray for the results of our method PROX, yellow for results without scene constraints, (3) rendering from a different view, that shows that the camera view is deceiving.

E_J	$E_{\mathcal{C}}$	$E_{\mathcal{P}}$	E_D	Run time	
✓	✗	✗	✗	33.75	sec
✓	✓	✗	✗	46.91	
✓	✗	✓	✗	42.68	
✓	✓	✓	✗	47.64	
✓	✗	✗	✓	54.28	
✓	✓	✓	✓	73.08	

Table 3.2: Runtime for all configurations of our approach.

3.3.3 Computational Complexity

Table 3.2 reports the average runtime for all our configurations for 10 randomly sampled frames. Compared to using RGB alone; PROX improved “V2V” by 24% with a runtime increase of 41%.

3.3.4 Choice of Contact Vertices

We choose the body vertices that often come in contact with the 3D world. This choice is not exclusive. Table 3.3 evaluates different sets of candidate contact vertices, namely our annotations and all vertices. Mean errors without Procrustes alignment, “PJE” and “V2V”, increase when all the vertices are used. In addition, runtime increases by ~ 7 seconds. This suggests the importance of affordances and semantics; future work can learn the likely contact vertices for different object classes in a data-driven fashion. To this end, the community first needs training data similar to the data generated by our work.

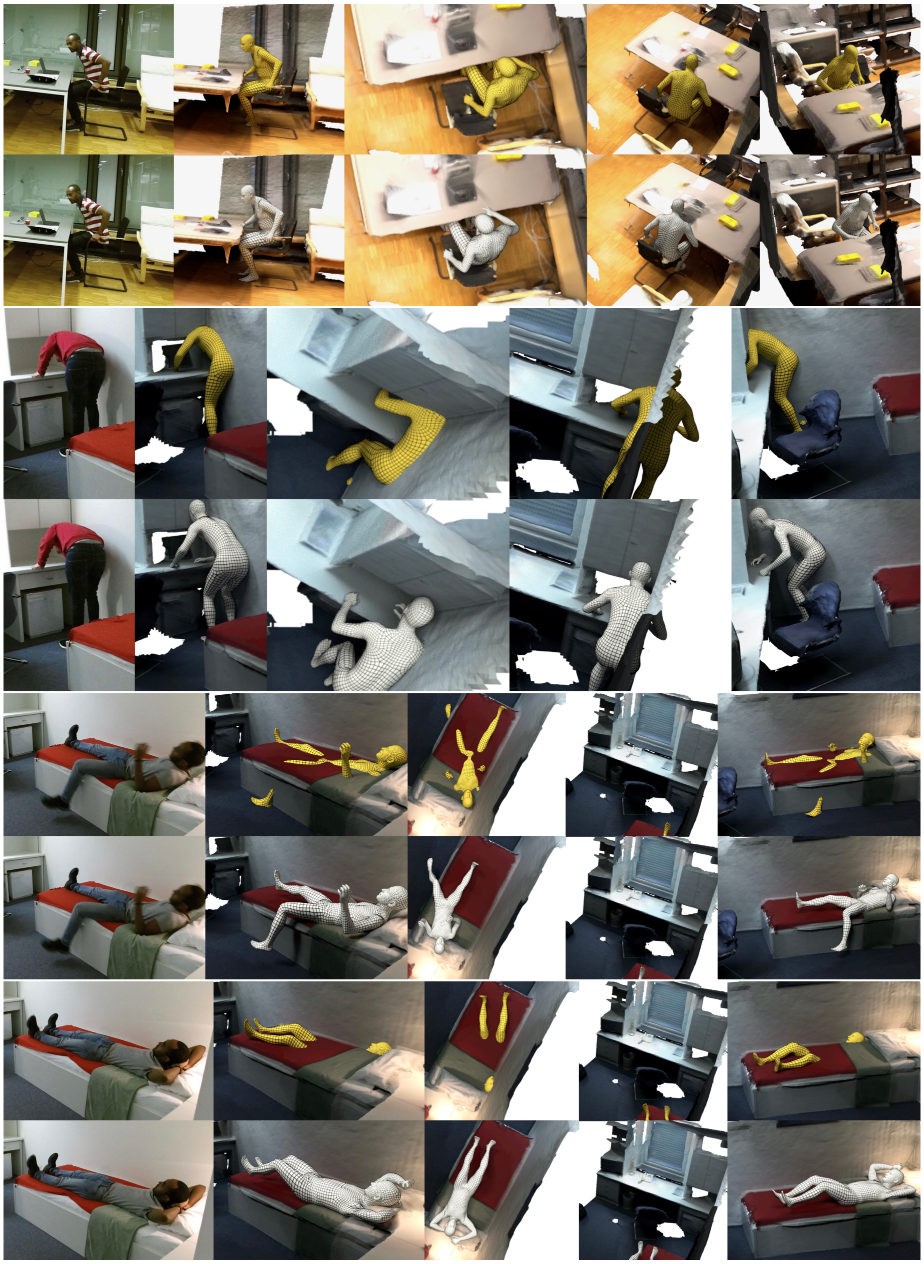


Figure 3.8: Qualitative results on our PROX dataset. The human body pose is estimated *with* (light gray) and *without* (yellow) our scene constraints. We show from left to right: (1) RGB images, (2) renderings from different viewpoints.

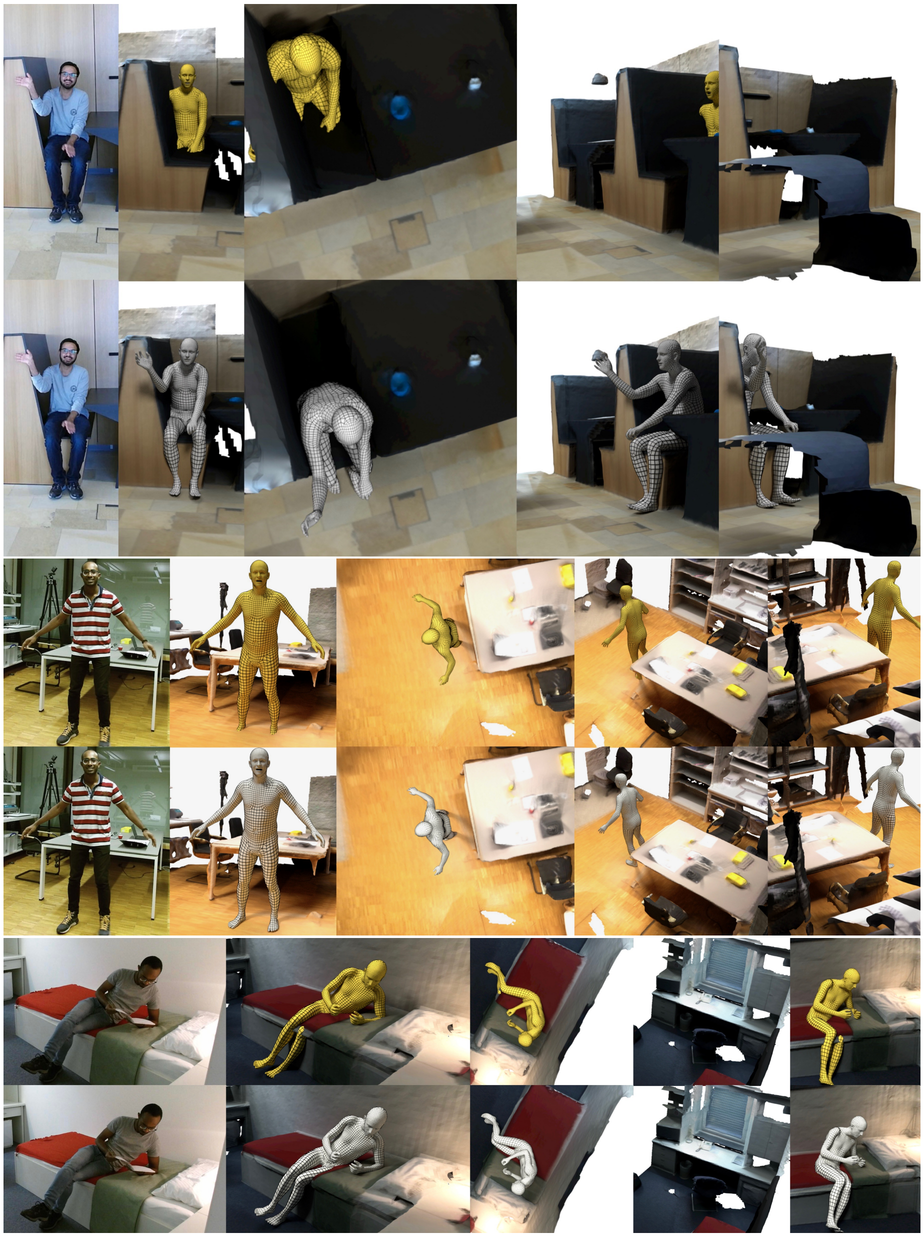


Figure 3.9: Qualitative results on our PROX dataset. The human body pose is estimated *with* (light gray) and *without* (yellow) our scene constraints. We show from left to right: (1) RGB images, (2) renderings from different viewpoints.

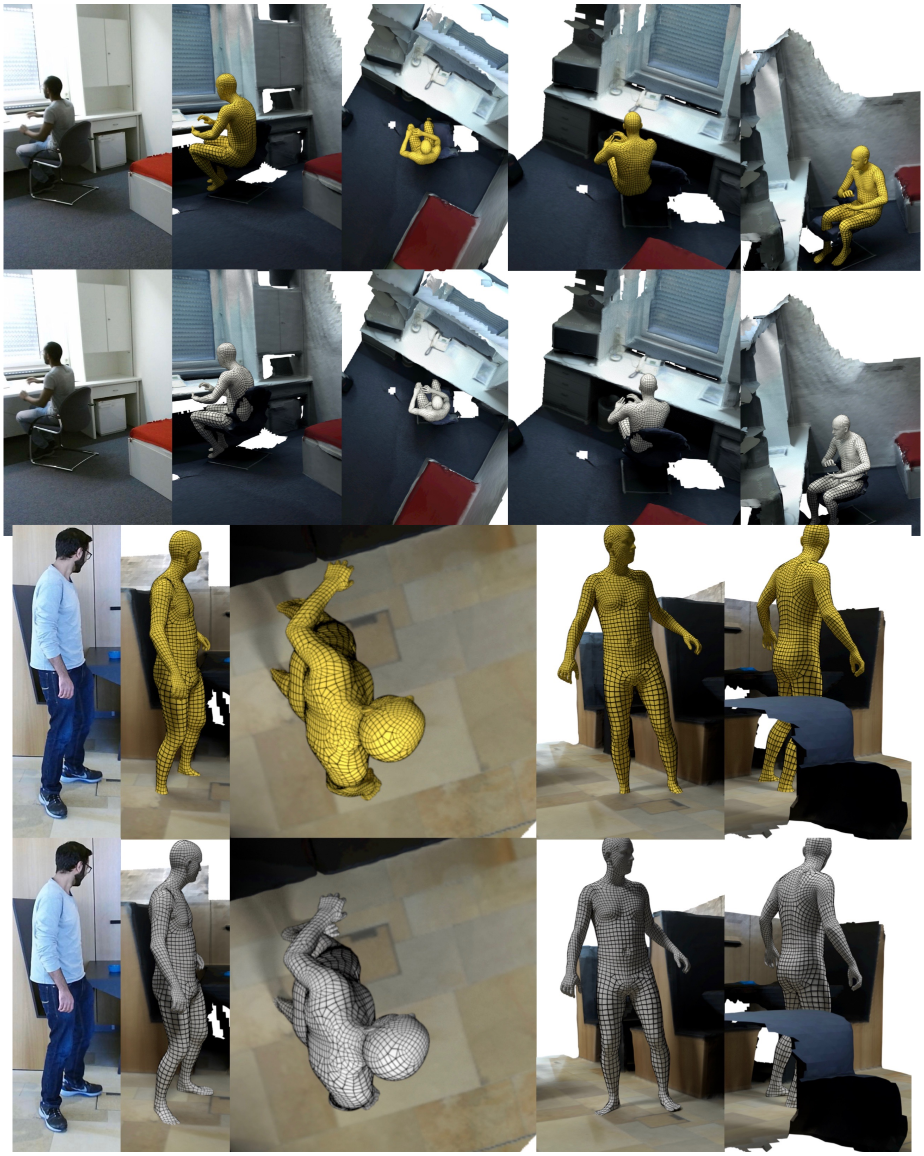


Figure 3.10: Qualitative results on our PROX dataset. The human body pose is estimated *with* (light gray) and *without* (yellow) our scene constraints. We show from left to right: (1) RGB images, (2) renderings from different viewpoints.

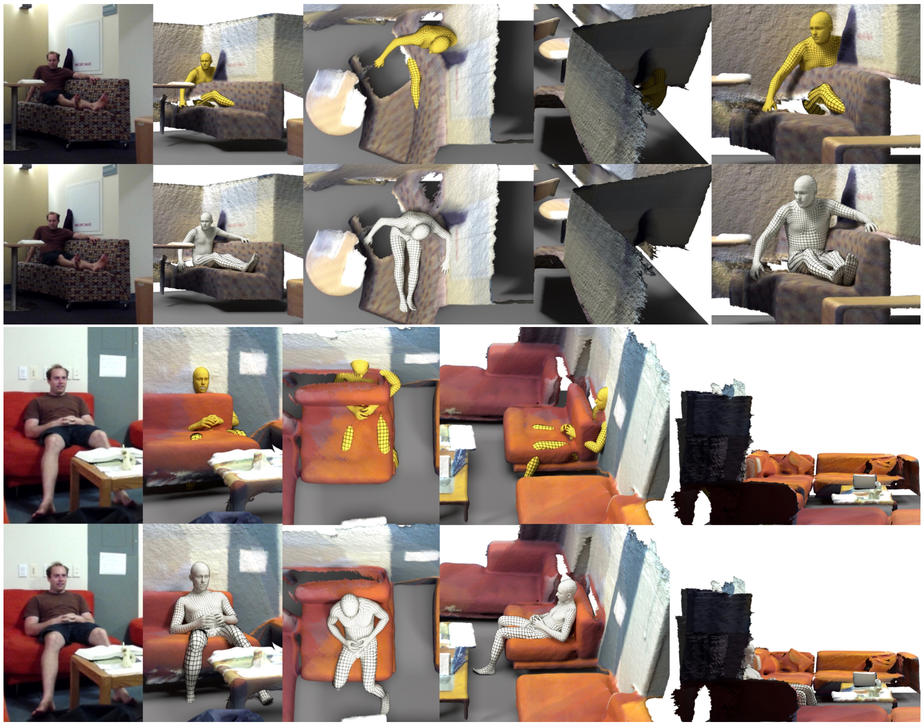


Figure 3.11: Qualitative results on the PiGraphs [177] dataset. The human body pose is estimated *with* (gray color) and *without* (yellow color) our environmental terms. Please note that [177] estimate just a 3D skeleton of only the major body joints. We show from left to right: (1) RGB images, (2) renderings from different viewpoints.

Contact vertices	PJE	V2V	p.PJE	p.V2V	
Selected of Fig. 3.2	208.03	208.57	72.76	60.95	mm
All selected	217.82	216.62	72.35	60.16	

Table 3.3: Different sets of candidate contact vertices.

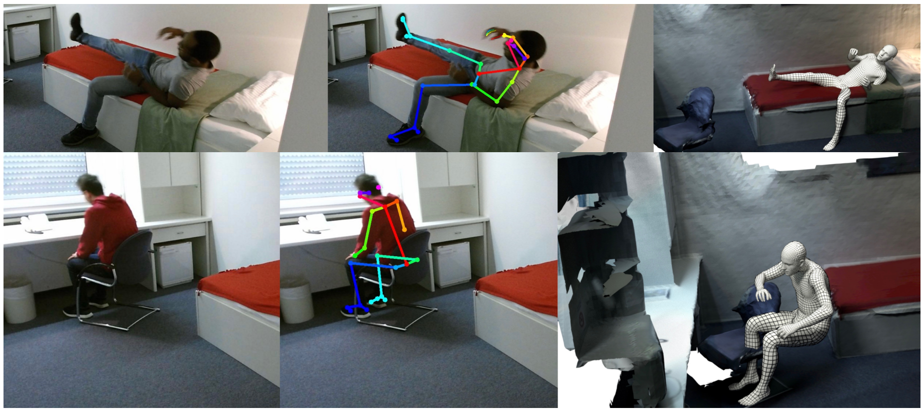


Figure 3.12: Representative failure cases on our PROX dataset. We show from left to right: (1) RGB image, (2) OpenPose result overlayed on the RGB image, (3) result of our method.

3.3.5 Failure Cases

Figures 3.12-3.13 show failure cases of our method (light gray) on our PROX dataset. For each example, we show from left to right: (1) RGB image, (2) OpenPose result overlayed on the RGB image, (3) result of our method. Figure 3.12-top shows that our method still results in some penetration. Our assumption of a static scene is not always true; in this case the bed is deformable and its shape changes during interaction. Future work should explore modeling deformations of the human body and the world. Figure 3.12-bottom shows a failure of our inter-penetration term. In cases where initialization of body translation is not accurate enough, the optimizer might end up in a local minimum that is not always in agreement with the real pose in 3D space. Figure 3.13 shows typical failure cases of OpenPose. In Figure 3.13-top, the left leg is not detected correctly, while in Figure 3.13-middle and Figure 3.13-bottom several body joints are flipped by OpenPose.

3.4 Discussion

Why such constraints are not typically used? One key reason is that to estimate and reason about contact and inter-penetration, one needs *both* a model of the 3D *scene* and a realistic model of the *human body*. The former is easy to obtain today with many scanning technologies but, if the body model is not accurate, it does not make sense to reason about contact and inter-penetration. Consequently, we use the SMPL-X body model [153], which is realistic enough to serve as a “proxy” for the real human in the 3D scene. In particular, the feet, hands, and body of the model have realistic shape and degrees of freedom.

Is it realistic to assume a 3D scene for refining pose? Here we assume that a rough 3D model of the scene is available; one could argue that this is a hard assump-

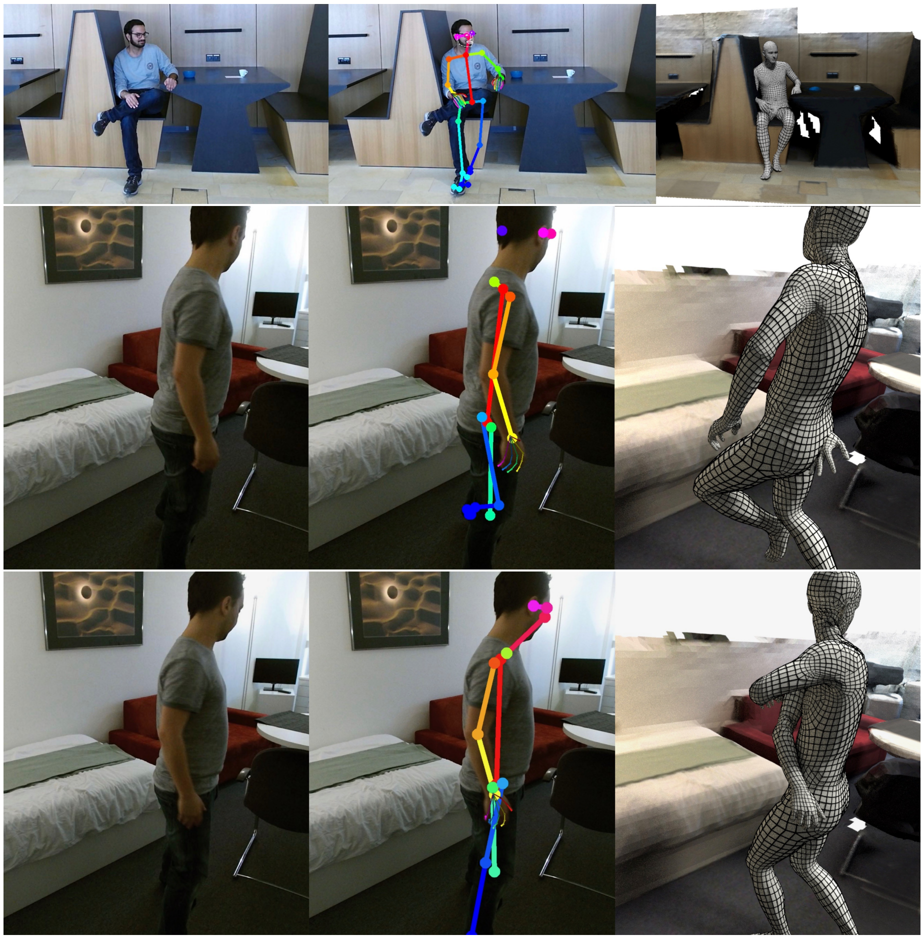


Figure 3.13: Representative failure cases on our PROX dataset. We show from left to right: (1) RGB image, (2) OpenPose result overlayed on the RGB image, (3) result of our method.

tion. Reconstructing a 3D scene from a single RGB image is a hot research topic, but the problem is ill-posed and currently unsolved. Here we want to show, in the first place, that knowledge about the scene helps pose estimation. Our results support this hypothesis, and scanning a scene today is quite easy. Future work can relax this assumption, and move to the more difficult problem of exploiting recent deep networks to estimate the scene directly from monocular RGB images. There are now good methods to infer depth maps from a single image [45] as well as methods that do more semantic analysis and estimate 3D CAD models of the objects in the scene [142]. Our work is complementary to this direction, and we believe that monocular 3D scene estimation and monocular 3D human pose estimation should happen together. The work here provides a clear example of why this is valuable.

3.5 Conclusion

In PROX we focus on human-world interactions and capture the motion of humans interacting with a real static 3D scene in RGB images. We use a holistic model, SMPL-X [153], that jointly models the body with face and fingers, which are important for interactions. We show that incorporating interaction-based human-world constraints in an optimization framework (PROX) results in significantly more realistic and accurate MoCap. We also collect a new dataset of 3D scenes with RGB-D sequences involving human interactions and occlusions. We perform extensive quantitative and qualitative evaluations that clearly show the benefits of incorporating scene constraints into 3D human pose estimation. Our code, data and MoCap are available for research purposes.

3.6 Limitations and Future work

A limitation of the current formulation is that we do not model scene occlusion. Current 2D part detectors do not indicate when joints are occluded and may provide inaccurate results. By knowing the scene structure, we could reason about what is visible and what is not. Another interesting direction would be the unification of the self-penetration and the body-scene inter-penetration by employing the implicit formulation of [193] for the whole body. Future work can exploit recent deep networks to estimate the scene directly from monocular RGB images. More interesting directions would be to extend our method to dynamic scenes [172], human-human interaction and to account for scene and body deformation.

Chapter 4

Populating 3D Scenes by Learning Human-Scene Interaction

Humans constantly interact with the world around them. We move by walking on the ground; we sleep lying on a bed; we rest sitting on a chair; we work using touchscreens and keyboards. Our bodies have evolved to exploit the affordances of the natural environment, and we design objects to better "afford" our bodies. While obvious, it is worth stating that these physical interactions involve contact. Despite the importance of such interactions, existing representations of the human body do not explicitly represent, support, or capture them.

In computer vision, human pose is typically estimated in isolation from the 3D scene, while in computer graphics 3D scenes are often scanned and reconstructed without people. Both the recovery of humans in scenes and the automated synthesis of realistic people in scenes remain challenging problems. Automation of this latter case would reduce animation costs and open up new applications in augmented reality. Here we take a step towards automating the realistic placement of 3D people in 3D scenes with realistic contact and semantic interactions (Fig. 4.1). We develop a novel *body-centric* approach that relates 3D body shape and pose to possible world interactions. Learned parametric 3D human models [12, 94, 123, 153] represent the shape and pose of people accurately. We employ the SMPL-X [153] model, which includes the hands and face, as it supports reasoning about contact between the body and the world.

While such body models are powerful, we make three *key observations*. First, human models like SMPL-X [153] do not explicitly model contact. Second, not all parts of the body surface are equally likely to be in contact with the scene. Third, the poses of our body and scene semantics are highly intertwined. Imagine a person sitting on a chair; body contact likely includes the buttocks, probably also the back, and maybe the arms. Think of someone opening a door; their feet are likely in contact with the floor, and their hand is in contact with the doorknob.

Based on these observations, we formulate a novel model, that makes human-scene

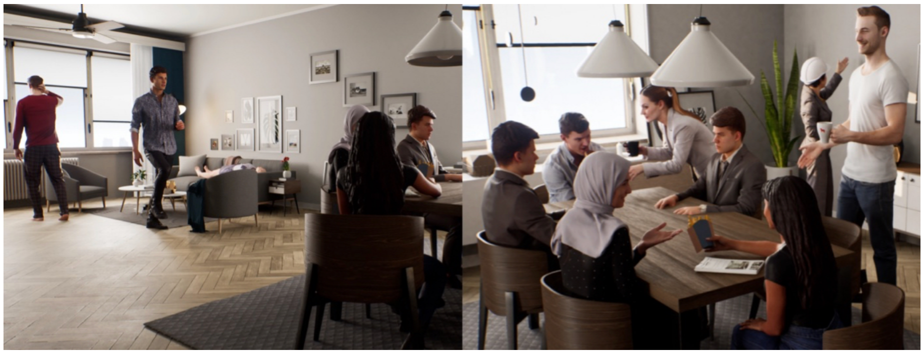


Figure 4.1: POSA automatically places 3D people in 3D scenes such that the interactions between the people and the scene are both geometrically and semantically correct. POSA exploits a new learned representation of human bodies that explicitly models how bodies interact with scenes.

interaction (HSI) an explicit and integral part of the body model. The key idea is to encode HSI in an egocentric representation built in SMPL-X. This effectively extends the SMPL-X model to capture contact and the semantics of HSI in a body-centric representation. We call this POSA for *"Pose with prOximitieS and contActs"*. Specifically, for every vertex on the body and every pose, POSA defines a probabilistic feature map that encodes the probability that the vertex is in contact with the world and the distribution of semantic labels associated with that contact.

POSA is a conditional Variational Auto-Encoder (cVAE), conditioned on SMPL-X vertex positions. We train on the PROX dataset [73], which contains 20 subjects, fit with SMPL-X meshes, interacting with 12 real 3D scenes. Please see Section. 3.2.1 for more details on the PROX dataset. We also train POSA using the scene semantic annotations provided by the PROX-E dataset [228]. Once trained, given a posed body, we can sample likely contacts and semantic labels for all vertices. We show the value of this representation with two challenging applications.

First, we focus on automatic scene population as illustrated in Fig. 4.1. That is, given a 3D scene and a body in a particular pose, where in the scene is this pose most likely? As demonstrated in Fig. 4.1 we use SMPL-X bodies fit to commercial 3D scans of people [150], and then, conditioned on the body, our cVAE generates a target POSA feature map. We then search over possible human placements while minimizing the discrepancy between the observed and target feature maps. We quantitatively compare our approach to PLACE [227], which is SOTA on a similar task, and find that POSA has higher perceptual realism.

Second, we use POSA for monocular 3D human pose estimation in a 3D scene. We build on the PROX method (Section. 3.1.4) that hand-codes contact points, and replace these with our learned feature map, which functions as an HSI prior. This automates a heuristic process, while producing lower pose estimation errors than the original PROX method.

To summarize, POSA is a novel model that intertwines SMPL-X pose and scene semantics with contact. To the best of our knowledge, this is the first learned human body model that incorporates HSI in the model. We think this is important because such a model can be used in all the same ways that models like SMPL-X are used but now with the addition of body-scene interaction. The key novelty is posing HSI as part of the body representation itself. Like the original learned body models, POSA provides a platform that people can build on. To facilitate this, our model and code are available for research purposes at https://posa.is.tue.mpg.de.

4.1 Method

4.1.1 Human Pose and Scene Representation

Our training data corpus is a set of n pairs of 3D meshes

$$\mathcal{M} = \{\{M_{b,1}, M_{s,1}\}, \{M_{b,2}, M_{s,2}\}, \ldots, \{M_{b,n}, M_{s,n}\}\}$$

comprising body meshes $M_{b,i}$ and scene meshes $M_{s,i}$. We drop the index, i, for simplicity when we discuss meshes in general. These meshes approximate human body surfaces $\mathcal{S}_b$ and scene surfaces $\mathcal{S}_s$. Scene meshes $M_s = (V_s, F_s, L_s)$ have a varying number of vertices $N_s = |V_s|$ and triangle connectivity F_s to model arbitrary scenes. They also have per-vertex semantic labels L_s. Human meshes are represented by SMPL-X [153]. Interested readers are referred to Section. 3.1.3 for more details about SMPL-X.

4.1.2 POSA Representation for HSI

We encode the relationship between the human mesh $M_b = (V_b, F_b)$ and the scene mesh $M_s = (V_s, F_s, L_s)$ in an egocentric feature map f that encodes per-vertex features on the SMPL-X mesh M_b. We define f as:

$$f : (V_b, M_s) \to [f_c, f_s] \,, \tag{4.1}$$

where f_c is the contact label and f_s is the semantic label of the contact point. N_f is the feature dimension.

For each vertex i on the body, V_b^i, we find its closest scene point $P_s = \text{argmin}_{P_s \in \mathcal{S}_s} \|P_s - V_b^i\|$. Then we compute the distance f_d:

$$f_d = \|P_s - V_b^i\| \in \mathbb{R}. \tag{4.2}$$

Given f_d, we can compute whether a V_b^i is in contact with the scene or not, with f_c:

$$f_c = \begin{cases} 1 & f_d \leq \textit{Contact Threshold}, \\ 0 & f_d > \textit{Contact Threshold}. \end{cases} \tag{4.3}$$

The contact threshold is chosen empirically to be 5 cm. The semantic label of the contacted surface f_s is a one-hot encoding of the object class:

$$f_s = \{0, 1\}^{N_o} \,, \tag{4.4}$$

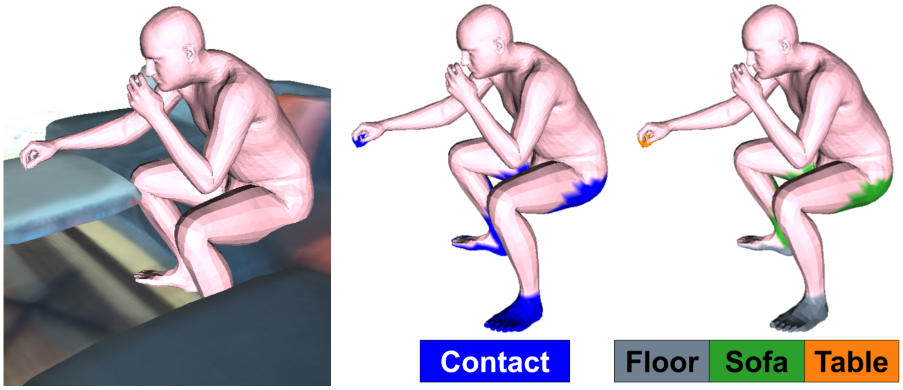


Figure 4.2: Illustration of our proposed representation. From left to right: An example of a SMPL-X mesh M_b in a scene M_s, with contact f_c, and scene semantics f_s on it. For f_c, blue means the body vertex is likely in contact. For f_s, the colors correspond to the scene semantic label.

where N_o is the number of object classes. The sizes of f_c, f_s, and f are 1, 40 and 41 respectively. All the features are computed once offline before training. A visualization of the proposed representation is in Fig. 4.2.

4.1.3 Learning

Our goal is to learn a probabilistic function from body pose and shape to the feature space of contact and semantics. That is, given a body, we want to sample labeling of the vertices corresponding to likely scene contacts and their corresponding semantic label. Note that this function, once learned, only takes the body as input and not a scene – it is a *body-centric* representation.

To train this, we use the PROX [73] dataset, which contains bodies in 3D scenes. We also use the scene semantic annotations from the PROX-E dataset [228]. For each body mesh M_b, we factor out the global translation and rotation R_y and R_z around the y and z axes. The rotation R_x around the x axis is essential for the model to differentiate between, e.g., standing up and lying down.

Given pose and shape parameters in a given training frame, we compute a $M_b = M(\theta, \beta, \psi)$. This gives vertices V_b from which we compute the feature map that encodes whether each V_b^i is in contact with the scene or not, and the semantic label of the scene contact point P_s.

We train a conditional Variational Autoencoder (cVAE), where we condition the feature map on the vertex positions, V_b, which are a function of the body pose and shape parameters. Training optimizes the encoder and decoder parameters to minimize

$\mathcal{L}_{\text{total}}$ using gradient descent:

$$\begin{aligned}
\mathcal{L}_{\text{total}} &= \alpha * \mathcal{L}_{KL} + \mathcal{L}_{rec}, &(4.5)\\
\mathcal{L}_{KL} &= KL(Q(z|f, V_b)||p(z)), &(4.6)\\
\mathcal{L}_{rec}\left(f, \hat{f}\right) &= \lambda_c * \sum_i \text{BCE}\left({f_c}^i, {\hat{f}_c}^i\right) \\
&\quad + \lambda_s * \sum_i \text{CCE}\left(f_s^i, {\hat{f}_s}^i\right), &(4.7)
\end{aligned}$$

where $\hat{f}_c$ and $\hat{f}_s$ are the reconstructed contact and semantic labels, *KL* denotes the Kullback Leibler divergence, and $\mathcal{L}_{rec}$ denotes the reconstruction loss. BCE and CCE are the binary and categorical cross entropies respectively. The α is a hyperparameter inspired by Gaussian β-VAEs [82], which regularizes the solution; here $\alpha = 0.05$. $\mathcal{L}_{rec}$ encourages the reconstructed samples to resemble the input, while $\mathcal{L}_{KL}$ encourages $Q(z|f, V_b)$ to match a prior distribution over z, which is Gaussian in our case. We set the values of λ_c and λ_s to 1.

Since f is defined on the vertices of the body mesh M_b, this enables us to use graph convolution as our building block for our VAE. Specifically, we use the Spiral Convolution formulation introduced in [21, 59]. The spiral convolution operator for a node i in the body mesh is defined as:

$$f_k^i = \gamma_k \left(\|_{j \in S(i,l)} f_{k-1}^j \right), \tag{4.8}$$

where γ_k denotes layer k in a multi-layer perceptron (MLP) network, and $\|$ is a concatenation operation of the features of neighboring nodes, $S(i, l)$. The spiral sequence $S(i, l)$ is an ordered set of l vertices around the central vertex i. Our architecture is shown in Fig. 4.3. For details on selecting and ordering vertices, please see [59].

4.1.4 Training Details

The global orientation of the body is typically irrelevant in our body-centric representation, so we rotate the training bodies around the y and z axes to put them in a canonical orientation. The rotation around the x axis, however, is essential to enable the model to differentiate between standing up and lying down. The semantic labels for the PROX scenes are taken from Zhang et al. [228], where scenes were manually labeled following the object categorization of Matterport3D [29], which incorporates 40 object categories.

Our encoder-decoder architecture is similar to the one introduced in Gong et al. [59]. The encoder consists of 3 spiral convolution layers interleaved with pooling layers $3 \times \{\text{Conv}\,(64) \rightarrow \text{Pool}\,(4)\} \rightarrow \text{FC}\,(512)$. Pool stands for a downsampling operation as in COMA [164], which is based on contracting vertices. FC is a fully connected layer, and the number in the bracket next to it denotes the number of units in that layer. We add 2 additional fully connected layers to predict the parameters of the latent code, with fully connected layers of 256 units each. The input to the encoder is a body mesh M_b where, for each vertex i, we concatenate V_b^i vertex positions, and f

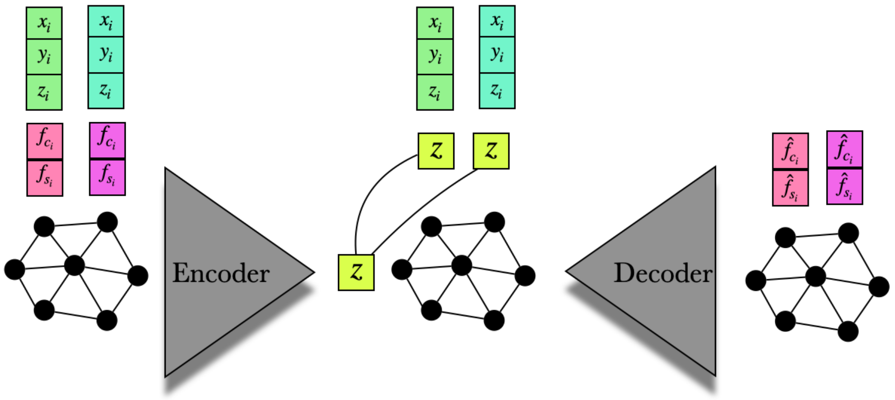


Figure 4.3: cVAE architecture. For each vertex on the body mesh, we concatenate the vertex positions x_i, y_i, z_i, the contact label f_c, and the corresponding semantic scene label f_s. The latent vector z is concatenated to the vertex positions, and the result passes to the decoder which reconstructs the input features $\hat{f}_c$, $\hat{f}_s$.

vertex features. For computational efficiency, we first downsample the input mesh by a factor of 4. So instead of working on the full mesh resolution of 10475 vertices, our input mesh has a resolution of 655 vertices. The decoder architecture consists of spiral convolution layers only $4 \times \{\text{Conv}\,(64)\} \rightarrow \text{Conv}\,(N_f)$. We attach the latent vector z to the 3D coordinates of each vertex similar to Kolotouros et al. [105].

We build our model using the PyTorch framework. We use the Adam optimizer [102], with a batch size of 64, and learning rate of $1e^{-3}$ without learning rate decay.

For computational efficiency, we employ a precomputed 3D signed distance field (SDF) for the static scene $\mathcal{S}_s$, as explained in Section 3.1.6. The SDF has a resolution of $512 \times 512 \times 512$. Each voxel c_j stores the distance $d_j \in \mathbb{R}$ of its centroid $P_j \in \mathbb{R}^3$ to the nearest surface point $P_s \in \mathcal{S}_s$. The distance d_j has a positive sign if P_j lies in the free space outside physical scene objects, while it has a negative sign if it is inside a scene object.

4.2 Experiments

We perform several experiments to investigate the effectiveness and usefulness of our proposed representation and model under different use cases, namely generating HSI features, automatically placing 3D people in scenes, and improving monocular 3D human pose estimation.

4.2.1 Random Sampling

We evaluate the generative power of our model by sampling different feature maps conditioned on novel poses using our trained decoder $P(f_{\text{Gen}}|z, V_b)$, where $z \sim \mathcal{N}(0, I)$ and f_{Gen} is the randomly generated feature map. This is equivalent to answering the question: "In this given pose, which vertices on the body are likely to be in contact with the scene, and what object would they contact?" Randomly generated samples are shown in Fig. 4.4.

We observe that our model generalizes well to various poses. For example, notice that when a person is standing with one hand pointing forward, our model predicts the feet and the hand to be in contact with the scene. It also predicts the feet are in contact with the floor and hand is in contact with the wall. However, this changes for the examples when a person is in a lying pose. In this case, most of the vertices from the back of the body are predicted to be in contact (blue color) with a bed (light purple) or a sofa (dark green). These features are predicted from the body alone; there is no notion of "action" here. Pose alone is a powerful predictor of interaction.

Since the model is probabilistic, we can sample many possible feature maps for a given pose. We show multiple randomly sampled feature maps for the same pose in Fig. 4.5. Note how POSA generates a variety of valid feature maps for the same pose. Notice for example that the feet are always correctly predicted to be in contact with the floor. Sometimes our model predicts the person is sitting on a chair (far left) or on a sofa (far right).

The predicted semantic map f_s is not always accurate as shown in the far right of Fig. 4.5. The model predicts the person to be sitting on a sofa but at the same time predicts the lower parts of the leg to be in contact with a bed which is unlikely.

From a dataset of 20 subjects only, our models learns to predict plausible feature maps for a wide range of human body shapes, as shown in Fig. 4.6.

4.2.2 Affordances: Putting People in Scenes

Given a posed 3D body and a 3D scene, can we place the body in the scene so that the pose makes sense in the context of the scene? That is, does the pose match the affordances of the scene [60, 100, 104]? Specifically, given a scene, M_s, semantic labels of objects present, and a body mesh, M_b, our method finds where in M_s this given pose is likely to happen. We solve this problem in two steps.

First, given the posed body, we use the decoder of our cVAE to generate a feature map by sampling $P(f_{\text{Gen}}|z, V_b)$ as in Sec. 4.2.1. Second, we optimize the objective function:

$$E(\gamma, \theta_0, \theta) = \mathcal{L}_{afford} + \mathcal{L}_{pen} + \mathcal{L}_{reg}, \tag{4.9}$$

where γ is the body translation, θ_0 is the global body orientation and θ is the body pose. The afforance loss $\mathcal{L}_{afford} =$

$$\lambda_1 * ||f_{\text{Gen}_c} \cdot f_d||_2^2 + \lambda_2 * \sum_i \text{CCE}\left(f_{\text{Gen}_s}^{\;i}, f_s^i\right), \tag{4.10}$$

f_d and f_s are the observed distance and semantic labels, which are computed using Eq. 4.2 and Eq. 4.4 respectively. f_{Gen_c} and f_{Gen_s} are the generated contact and semantic labels, and $\cdot$ denotes dot product. λ_1 and λ_2 are 1 and 0.01 respectively. $\mathcal{L}_{pen}$ is a penetration penalty to discourage the body from penetrating the scene:

$$\mathcal{L}_{pen} = \lambda_{pen} * \sum_{f_d^i < 0} \left(f_d^i\right)^2. \tag{4.11}$$

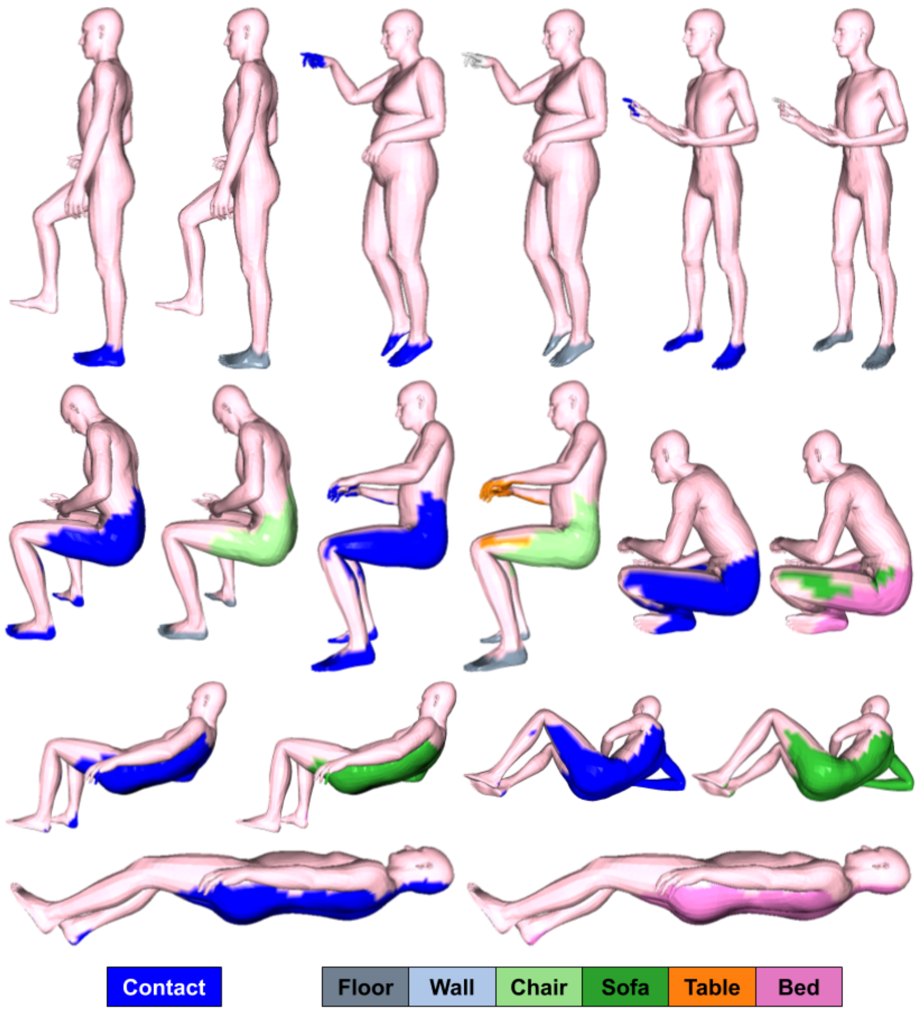


Figure 4.4: Random samples from our trained cVAE. For each example (image pair) we show from left to right: f_c and f_s. The color code is at the bottom. For f_c, blue means contact, while pink means no contact. For f_s, each scene category has a different color.

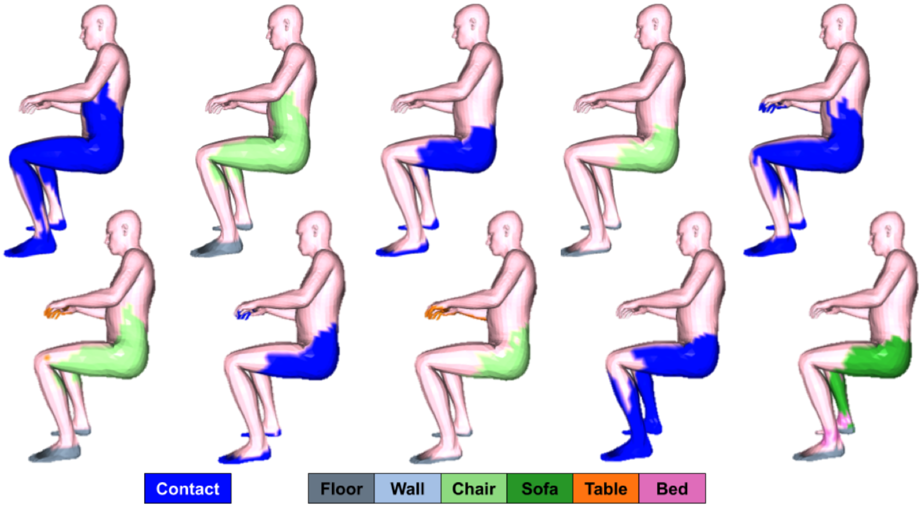


Figure 4.5: Random samples from our trained cVAE for the same pose. For each example, we show from left to right: f_c and f_s. The color code is at the bottom. For f_c, blue means contact, while pink means no contact. For f_s, each scene category has a different color.

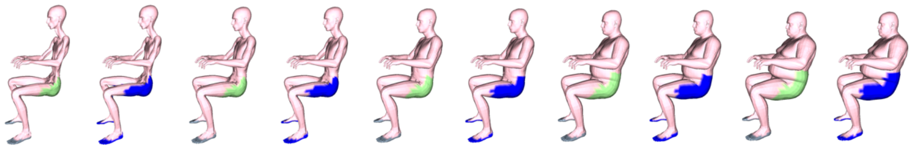


Figure 4.6: Generated feature maps for various body shapes.

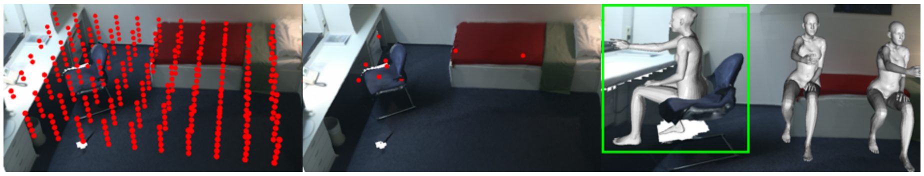


Figure 4.7: Main steps of our method for scene population. **(1)** Grid with all candidate positions. **(2)** The 10 best positions. **(3)** Final result.

$\lambda_{pen} = 10$. $\mathcal{L}_{reg}$ is a regularizer that encourages the estimated pose to remain close to the initial pose θ_{init} of M_b:

$$\mathcal{L}_{reg} = \lambda_{reg} * ||\theta - \theta_{\text{init}}||_2^2. \tag{4.12}$$

Although the body pose is given, we optimize over it, allowing the θ parameters to change slightly since the given pose θ_{init} might not be well-supported by the scene. This allows for small pose adjustment that might be necessary to better fit the body into the scene. $\lambda_{reg} = 100$.

The input posed mesh, M_b, can come from any source. For example, we can generate random SMPL-X meshes using VPoser [153] which is a VAE trained on a large dataset of human poses. More interestingly, we use SMPL-X meshes fit to realistic Renderpeople scans [150] (see Fig. 4.1).

In Fig. 4.7 we show the three main steps to populate a scene: **(1)** Given a scene, we create a regular grid of candidate positions (Fig. 4.7 (1)). We place the body, in a given pose, at each candidate position and evaluate Eq. 4.9 once. **(2)** We then keep the 10 best candidates with the lowest energy (Fig. 4.7 (2)), and **(3)** iteratively optimize Eq. 4.9 for these; Fig. 4.7 (3) shows results at three positions, with the best one highlighted in green.

We tested our method with both real (scanned) and synthetic (artist generated) scenes. Example bodies optimized to fit in a real scene from the PROX [73] test set are shown in Fig. 4.8 (top); this scene was not used during training. Note that people appear to be interacting naturally with the scene; that is, their pose matches the scene context. Figure 4.8 (bottom) shows bodies automatically placed in an artist-designed scene (Archviz Interior Rendering Sample, Epic Games)[1]. POSA goes beyond previous work [60, 100, 228] to produce realistic human-scene interactions for a wide range of poses like lying down and reaching out.

We show additional qualitative examples of SMPL-X meshes automatically placed in real and synthetic scenes in Fig. 4.9.

While the poses look natural in the above results, the SMPL-X bodies look out of place in realistic scenes. Consequently, we would like to render realistic people instead, but models like SMPL-X do not support realistic clothing and textures. In contrast, scans from companies like Renderpeople (Renderpeople GmbH, Köln) are realistic, but have a different mesh topology for every scan. The consistent topology of a mesh like SMPL-X is critical to learn the feature model.

[1] https://docs.unrealengine.com/en-US/Resources/Showcases/ArchVisInterior/index.html

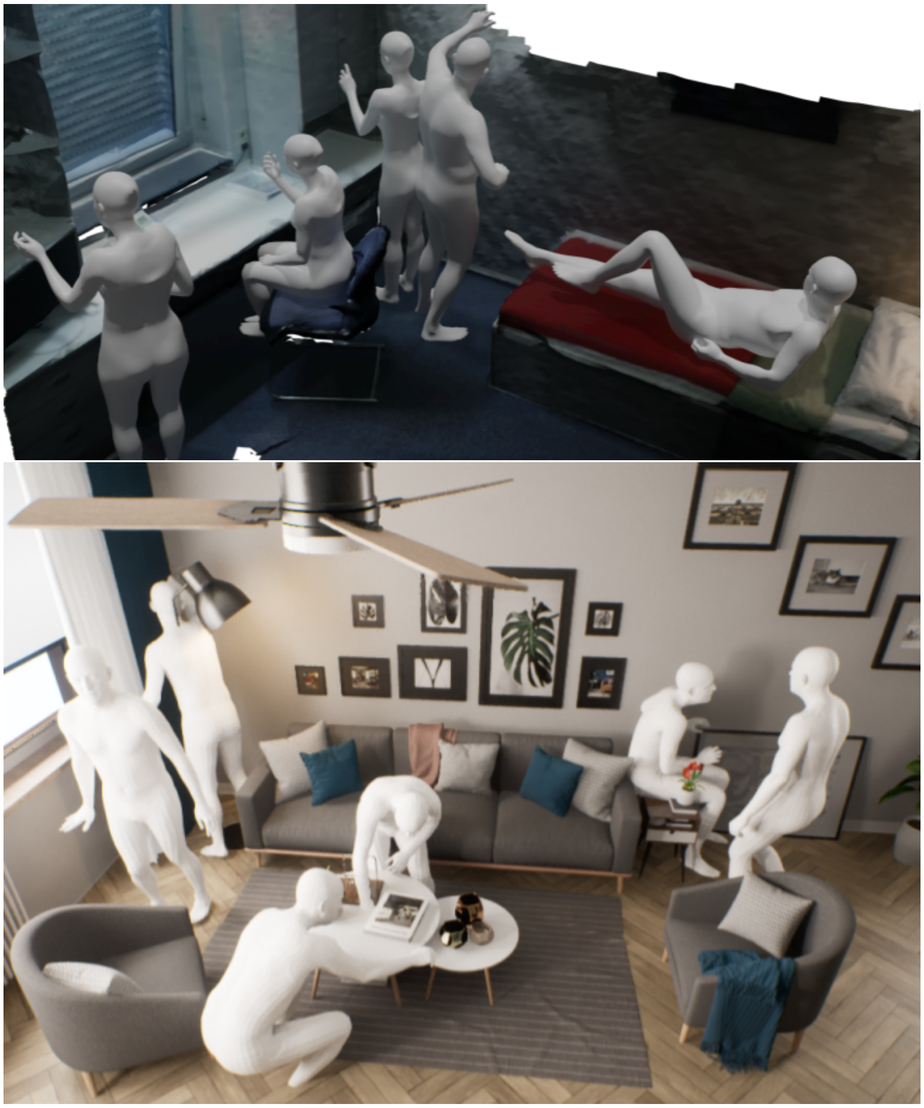


Figure 4.8: **(Top):** SMPL-X meshes automatically placed in a real scene from the PROX test set. The body shapes and poses here are drawn from the PROX test set and were not used in training. **(Bottom):** SMPL-X meshes automatically placed in a synthetic scene.

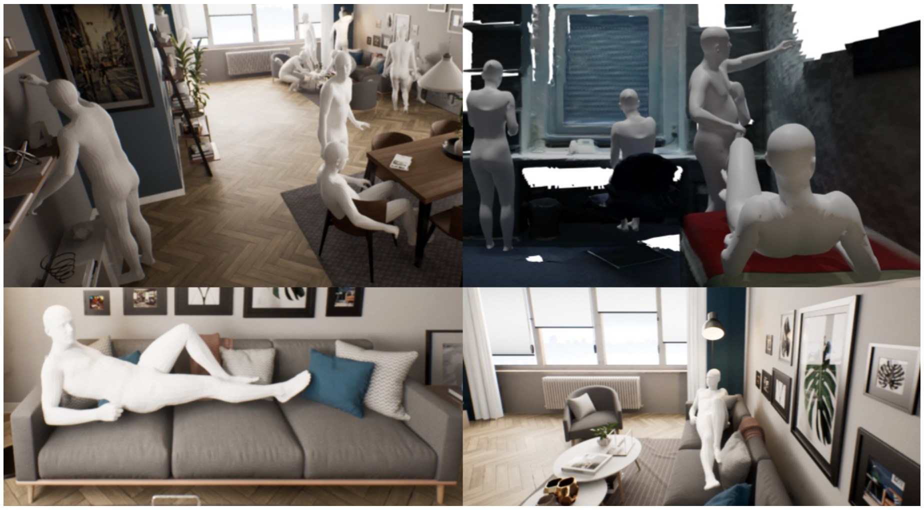


Figure 4.9: Qualitative examples of SMPL-X meshes automatically placed in real and synthetic scenes. The body shapes and poses were not used in training.

Clothed Humans: We address this issue by using SMPL-X fits to clothed meshes from the AGORA dataset [150]. We then take the SMPL-X fits and minimize an energy function similar to Eq. 4.9 with one important change. We keep the pose, θ, fixed:

$$E(\gamma, \theta_0) = \mathcal{L}_{afford} + \mathcal{L}_{pen}. \tag{4.13}$$

Since the pose does not change, we just replace the SMPL-X mesh with the original clothed mesh after the optimization converges.

The complete pipeline of the affordance detection task is shown in Fig. 4.10. Given a clothed 3D mesh that we want to put in a scene, we first need a SMPL-X fit to the mesh; here we take this from the AGORA dataset [150]. Then we generate a feature map using the decoder of our cVAE by sampling $P(f_{\text{Gen}}|z, V_b)$. Next we minimize the energy function in Eq. 4.13. Finally, we replace the SMPL-X mesh with the original clothed mesh.

Qualitative results for real scenes (Replica dataset [190]) are shown in Fig. 4.11, and for a synthetic scene in Fig. 4.1. Additional examples are shown in Fig. 4.12.

We show qualitative comparison between our results and PLACE [227] in Fig. 4.13. Note how our method generates more realistic and natural HSI.

Evaluation

We quantiatively evaluate POSA with two perceptual studies. In both, subjects are shown a pair of two rendered scenes, and must choose the one that best answers the question "Which one of the two examples has more realistic (i.e. natural and physically plausible) human-scene interaction?" We also evaluate physical plausibility and diversity.

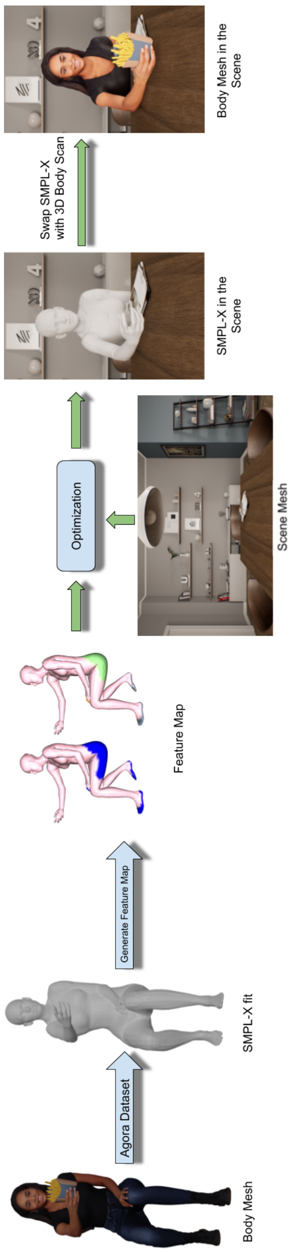


Figure 4.10: Putting realistic people in scenes. Pipeline of affordance detection using meshes with clothing. SMPL-X acts as a proxy for the clothed scan. POSA is used to sample features for this pose. These features are then used with the scene mesh to optimize the placement of the body. After convergence, we simply replace SMPL-X with the clothed scan.

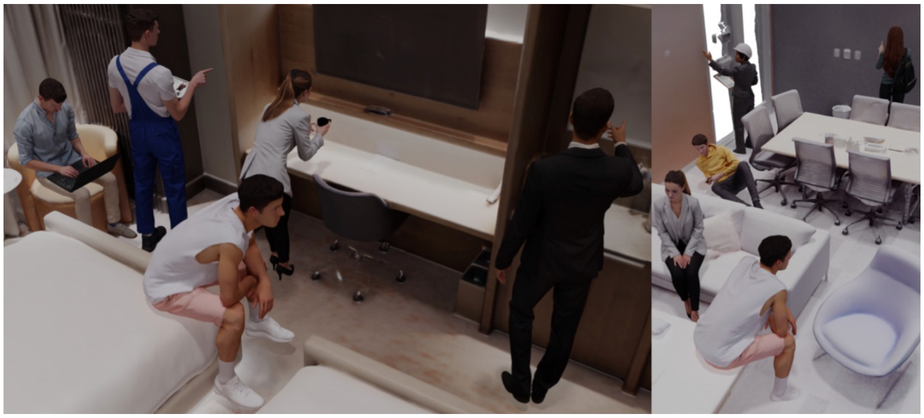


Figure 4.11: Unmodified clothed bodies (from Renderpeople) automatically placed in real scenes from the Replica dataset.

Comparison to PROX ground truth: We follow the protocol of Zhang et al. [227] and compare our results to randomly selected examples from PROX ground truth. We take 4 real scenes from the PROX [73] test set, namely MPH16, MPH1Library, N0SittingBooth and N3OpenArea. We take 100 SMPL-X bodies from the AGORA [150] dataset, corresponding to 100 different 3D scans from Renderpeople. We take each of these bodies and sample one feature map for each, using our cVAE. We then automatically optimize the placement of each sample in all the scenes, one body per scene. For unclothed bodies (Tab. 4.1, rows 1-3), this optimization changes the pose slightly to fit the scene (Eq. 4.9). For clothed bodies (Tab. 4.1, rows 4-5), the pose is kept fixed (Eq. 4.13). For each variant, this optimization results in 400 unique body-scene pairs. We render each 3D human-scene interaction from 2 views so that subjects are able to get a good sense of the 3D relationships from the images. Using Amazon Mechanical Turk (AMT), we show these results to 3 different subjects. This results in 1200 unique ratings. The results are shown in Tab. 4.1. The proposed POSA (contact + semantics) (row 3) outperforms both POSA (contact only) (row 2) and PLACE [227] (row 1), thus modeling scene semantics increases realism. Lastly, the rendering of high quality clothed meshes (bottom two rows) influences the perceived realism significantly.

Comparison between POSA and PLACE: We follow the same protocol as above, but this time we directly compare POSA and PLACE. The results are shown in Tab. 4.2. Again, we find that adding semantics improves realism. There are likely several reasons that POSA is judged more realistic than PLACE. First, POSA employs denser contact information across the whole SMPL-X body surface, compared to PLACE's sparse distance information through its basis point sets representation [162]. Second, POSA uses a human-centric formulation, as opposed to PLACE's scene-centric one, and this can induce better generalization across scenes. Third, POSA uses semantic features that help bodies do the right thing in the scene, while PLACE does not. When human generation is imperfect, inappropriate semantics may make the result seem worse. Fourth,

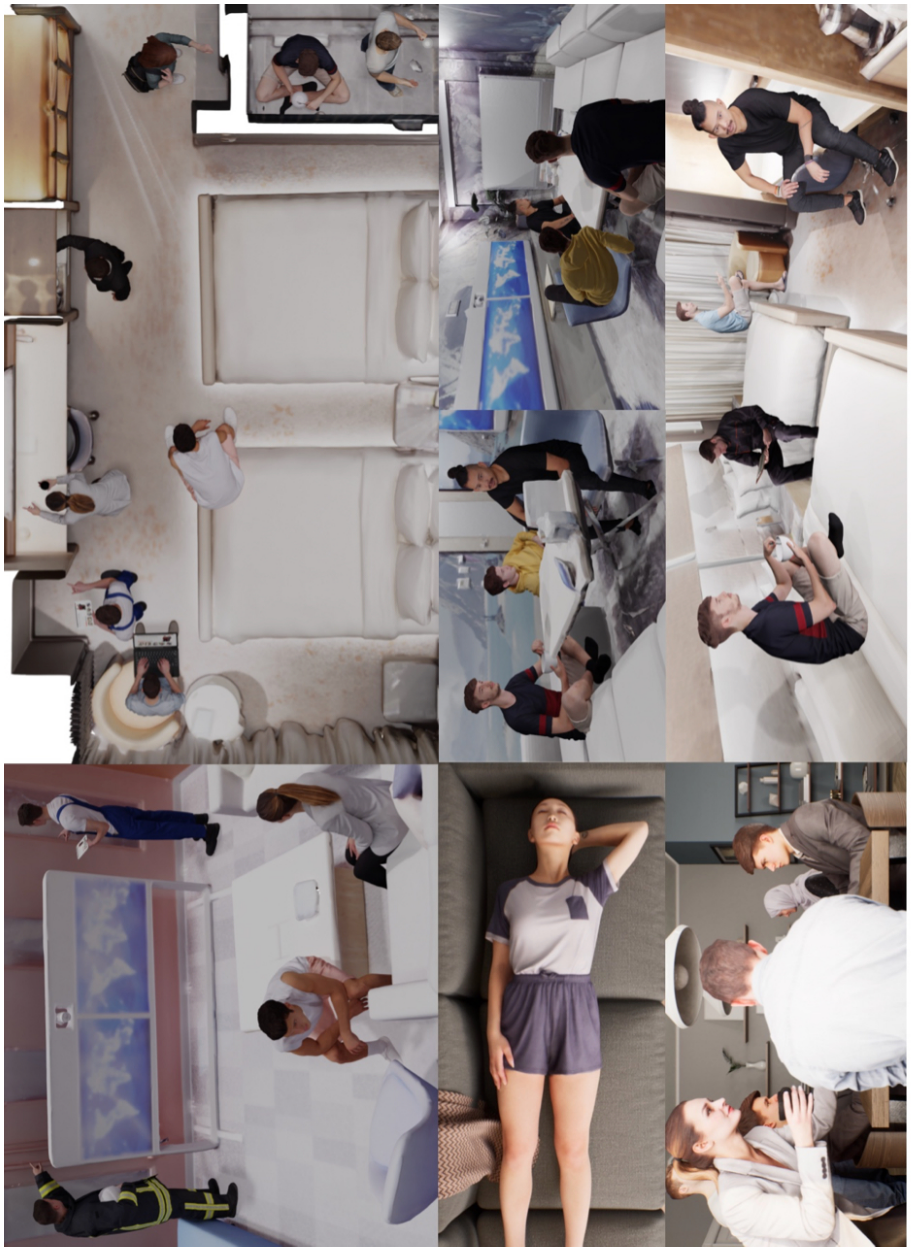


Figure 4.12: Clothed bodies (from Renderpeople) automatically placed in real and synthetic scenes.

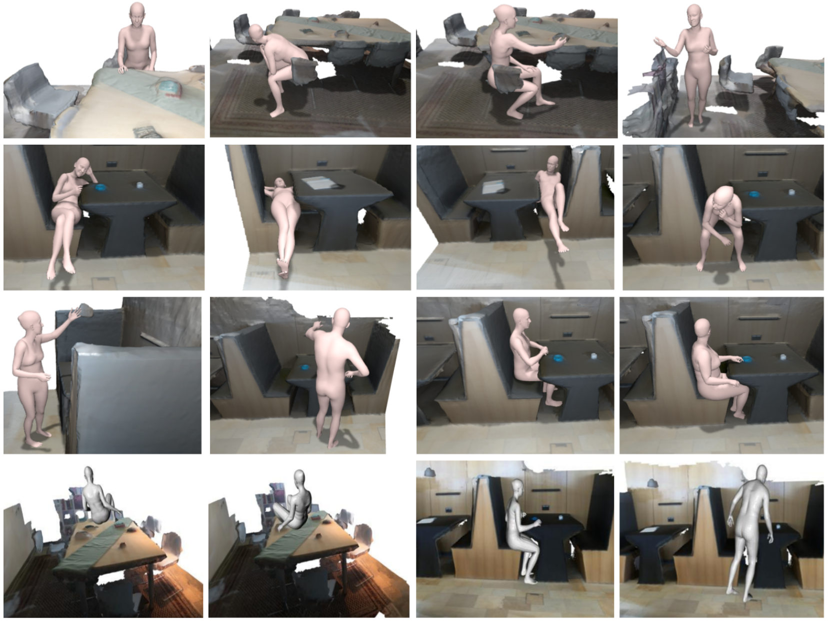


Figure 4.13: Qualitative examples from POSA (pink) and PLACE [227] (silver).

	Generation ↑	PROX GT ↓
PLACE [227]	48.5%	51.5%
POSA (contact only)	46.9%	53.1%
POSA (contact + semantics)	**49.1**%	50.1%
POSA-*clothing* (contact)	55.0%	45.0%
POSA-*clothing* (semantics)	**60.6**%	39.4%

Table 4.1: Comparison to PROX [73] ground truth. Subjects are shown pairs of a generated 3D human-scene interaction and PROX ground truth (GT), and must chose the most realistic one. A higher percentage means that subjects deemed this method more realistic.

	POSA-variant ↑	PLACE ↓
POSA (contact only)	60.7%	39.3%
POSA (contact + semantics)	**61.0**%	39.0%

Table 4.2: POSA compared to PLACE for 3D human-scene interaction generation. See Tab. 4.1 caption.

	Non-Collision ↑	Contact ↑
PSI [228]	0.94	0.99
PLACE [227]	**0.98**	0.99
POSA (contact only)	0.97	**1.0**
POSA (contact + semantics)	0.97	0.99

Table 4.3: Evaluation of the physical plausibility metric. Arrows indicate that higher scores are better.

	Entropy ↑	Cluster Size ↑
PSI [228]	**2.97**	2.53
PLACE [227]	2.91	**2.72**
POSA (contact only)	2.94	2.28
POSA (contact + semantics)	2.92	2.27

Table 4.4: Evaluation of the diversity metric. Arrows indicate that higher scores are better.

the two methods are solving slightly different tasks. PLACE generates a posed body mesh for a given scene, while our method samples one from the AGORA dataset and places it in the scene using a generated POSA feature map. While this gives PLACE an advantage, because it can generate an appropriate pose for the scene, it also means that it could generate an unnatural pose, hurting realism. In our case, the poses are always "valid" by construction, but may not be appropriate for the scene. Note that, while more realistic than prior work, the results are not always fully natural; sometimes people sit in strange places or lie where they usually would not.

Physical Plausibility: We take 1200 bodies from the AGORA [150] dataset and place all of them in each of the 4 test scenes of PROX, leading to a total of 4800 samples, following [227, 228]. Given a generated body mesh, M_b, a scene mesh, M_s, and a scene signed distance field (SDF) that stores distances d_j for each voxel j, we compute the following scores, defined by Zhang et al. [228]: (1) the *non-collision score* for each M_b, which is the ratio of body mesh vertices with positive SDF values divided by the total number of SMPL-X vertices, and (2) the *contact score* for each M_b, which is 1 if at least one vertex of M_b has a non-positive value. We report the mean non-collision score and mean contact score over all 4800 samples in Tab. 4.3; higher values are better for both metrics. POSA and PLACE are comparable under these metrics.

Diversity Metric: Using the same 4800 samples, we compute the diversity metric from [228]. We perform K-means ($k = 20$) clustering of the SMPL-X parameters of all sampled poses, and report: (1) the entropy of the cluster sizes, and (2) the cluster size, i.e. the average distance between the cluster center and the samples belonging to it. See Tab. 4.4; higher values are better. While PLACE generates poses and POSA samples them from a database, there is little difference in diversity.

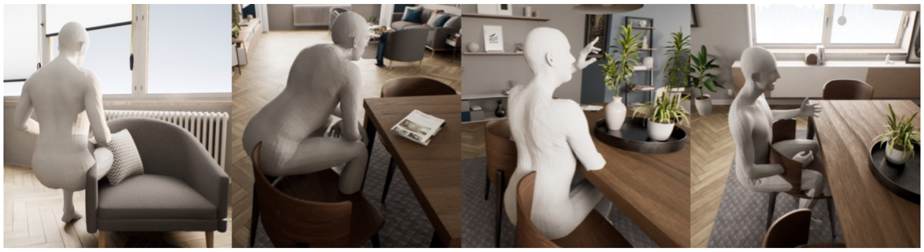


Figure 4.14: Failure cases.

Failure Cases

We show representative failure cases in Fig. 4.14. A common failure mode is residual penetrations; even with the penetration penalty, the body can still penetrate the scene. This can happen due to thin surfaces that are not captured by our SDF and/or because the optimization becomes stuck in a local minimum. In other cases, the feature map might not be right. This can happen when the model does not generalize well to test poses due to the limited training data.

4.2.3 Monocular Pose Estimation with HSI

Traditionally, monocular pose estimation methods focus only on the body and ignore the scene. Hence, they tend to generate bodies that are inconsistent with the scene. Here, we compare directly with PROX [73], which adds contact and penetration constraints to the pose estimation formulation. The contact constraint snaps a fixed set of contact points on the body surface to the scene, if they are "close enough". In PROX, however, these contact points are manually selected and are independent of pose.

We replace the hand-crafted contact points of PROX with our learned feature map. We fit SMPL-X to RGB image features such that the contacts are consistent with the 3D scene and its semantics. Similar to PROX, we build on SMPLify-X [153]. Specifically, SMPLify-X optimizes SMPL-X parameters to minimize an objective function of multiple terms: the re-projection error of 2D joints, priors and physical constraints on the body; $E_{\text{SMPLify-X}}(\beta, \theta, \psi, \gamma) =$

$$E_J + \lambda_\theta E_\theta + \lambda_\alpha E_\alpha + \lambda_\beta E_\beta + \lambda_{\mathcal{P}} E_{\mathcal{P}} \tag{4.14}$$

where θ represents the pose parameters of the body, face (neck, jaw) and the two hands, $\theta = \{\theta_b, \theta_f, \theta_h\}$, γ denotes the body translation, and β the body shape. E_J is a re-projection loss that minimizes the difference between 2D joints estimated from the RGB image I and the 2D projection of the corresponding posed 3D joints of SMPL-X. $E_\alpha(\theta_b) = \sum_{i \in (elbows, knees)} \exp(\theta_i)$ is a prior penalizing extreme bending only for elbows and knees. The term $E_{\mathcal{P}}$ penalizes self-penetrations. For more details please see [153] and Section. 3.1.4.

We turn off the PROX contact term and optimize Eq. 4.14 to get a pose matching the image observations and roughly obeying scene constraints. Given this rough body pose,

(mm)	PJE ↓	V2V ↓	p.PJE ↓	p.V2V ↓
RGB	220.27	218.06	73.24	60.80
PROX	167.08	166.51	**71.97**	**61.14**
POSA	**154.33**	**154.84**	73.17	63.23

Table 4.5: Pose estimation results for PROX and POSA. *PJE* is the mean per-joint error and *V2V* is the mean vertex-to-vertex Euclidean distance between meshes (after only pelvis joint alignment). The prefix "p" means that the error is computed after Procrustes alignment to the ground truth; this hides many errors, making the methods comparable.

which is not expected to change significantly, we sample features from $P(f_{\text{Gen}}|z, V_b)$ and keep these fixed. Finally, we refine by minimizing $E(\beta, \theta, \psi, \gamma, M_s) =$

$$E_{\text{SMPLify-X}} + ||f_{\text{Gen}_c} \cdot f_d|| + \mathcal{L}_{pen} \tag{4.15}$$

where $E_{\text{SMPLify-X}}$ represents the SMPLify-X energy term as defined in Eq. 4.14, f_{Gen_c} are the generated contact labels, f_d is the observed distance, and $\mathcal{L}_{pen}$ represents the body-scene penetration loss as in Eq. 4.11. We compare our results to standard PROX in Tab. 4.5. We also show the results of RGB-only baseline introduced in PROX for reference. Using our learned feature map improves accuracy over the PROX's heuristically determined contact constraints.

4.3 Conclusions

Traditional 3D body models, like SMPL-X, model the a priori probability of possible body shapes and poses. We argue that human poses in isolation from the scene, make little sense. We introduce POSA, which effectively upgrades a 3D human body model to explicitly represent possible human-scene interactions. Our novel, body-centric, representation encodes the contact and semantic relationships between the body and the scene. We show that this is useful and supports new tasks. For example, we consider placing a 3D human into a 3D scene. Given a scan of a person with a known pose, POSA allows us to search the scene for locations where the pose is likely. This enables us to populate empty 3D scenes with higher realism than the state of the art. We also show that POSA can be used for estimating human pose from an RGB image, and that the body-centered HSI representation improves accuracy. In summary, POSA is a good step towards a richer model of human bodies that goes beyond pose to support the modeling of HSI.

Limitations: POSA requires an accurate scene SDF; a noisy scene mesh can lead to penetration between the body and scene. POSA focuses on a single body mesh only. Penetration between clothing and the scene is not handled and multiple bodies are not considered. Optimizing the placement of people in scenes is sensitive to initialization and is prone to local minima. A simple user interface would address this, letting naive users roughly place bodies, and then POSA would automatically refine the placement.

Chapter 5

Stochastic Scene-Aware Motion Prediction

The computer vision community has made substantial progress on 3D scene understanding and on capturing 3D human motion, but less work has focused on synthesizing 3D people in 3D scenes. The advances in these two sub-fields, however, have provided tools for, and have created interest in, embodied agents for virtual worlds (e.g. [129, 179, 214, 215]) and in placing humans into scenes (e.g. [25, 76]). Creating virtual humans that move and act like real people, however, is challenging and requires tackling many smaller but difficult problems such as perception of unseen environments, plausible human motion modeling, and embodied interaction with complex scenes. While advances have been made in human locomotion modeling [85, 119] thanks to the availability of large scale datasets [27, 125, 141, 182, 196], realistically synthesizing virtual humans moving and interacting with 3D scenes, remains largely unsolved.

Imagine instructing a virtual human to "sit on a couch" in a cluttered scene, as illustrated in Fig. 5.1. To achieve this goal, the character needs to perform a series of complex actions. First, it should navigate through the scene to reach the target object while avoiding collisions with other objects in the scene. Next, the character needs to choose a *contact point* on the couch that will result in a plausible sitting action facing the right direction. Finally, if the character performs this action multiple times, there should be natural variations in the motion, mimicking real-world human-scene interactions; e.g., sitting on different parts of the couch with different styles such as with crossed legs, arms in different poses, etc. Achieving these goals requires a system to jointly reason about the scene geometry, smoothly transition between cyclic (e.g., walking) and acyclic (e.g., sitting) motions, and to model the diversity of human-scene interactions.

To this end, we propose SAMP for *Scene-Aware Motion Prediction*. SAMP is a stochastic model that takes a 3D scene as input, samples valid interaction goals, and generates goal-conditioned and scene-aware motion sequences of a character depicting realistic dynamic character-scene interactions. At the core of SAMP is a novel

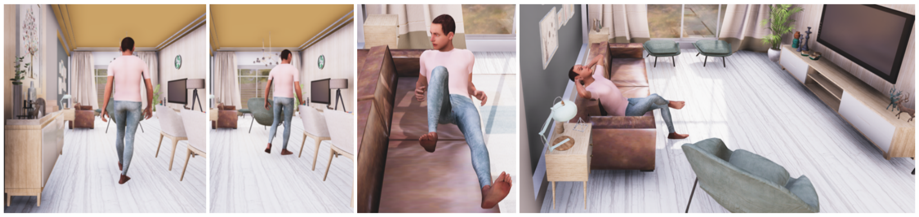


Figure 5.1: SAMP synthesizes virtual humans navigating complex scenes with realistic and diverse human-scene interactions.

autoregressive conditional variational autoencoder (cVAE) called MotionNet. Given a target object and an action, MotionNet samples a random latent vector at each frame to condition the next pose both on the previous pose of the character as well as the random vector. This enables MotionNet to model a wide range of styles while performing the target action. Given the geometry of the target object, SAMP further uses another novel neural network called GoalNet to generate multiple plausible contact points and orientations on the target object (e.g., different positions and sitting orientations on the cushions of a sofa). This component enables SAMP to generalize across objects with diverse geometry. Finally, to ensure the character avoids obstacles while reaching the goal in a cluttered scene, we use an explicit path planning algorithm (A* search) to pre-compute an obstacle-free path between the starting location of the character and the goal. This piecewise linear path consists of multiple way-points, which SAMP treats as intermediate goals to drive the character around the scene. SAMP runs in real-time at 30 fps. To the best of our knowledge, these individual components make SAMP the first system that addresses the problem of generating diverse dynamic motion sequences that depict realistic human-scene interactions in cluttered environments.

Training SAMP requires a dataset of rich and diverse character scene interactions. Existing large-scale MoCap datasets are largely dominated by locomotion and the few interaction examples lack diversity. Additionally, traditional MoCap focuses on the body and rarely captures the scene. Hence, we capture a new dataset covering various human-scene interactions with multiple objects. In each motion sequence, we track both the body motion and the object using a high resolution optical marker MoCap system. The dataset is available for research purposes.

Our contributions are: (1) A novel stochastic model for synthesizing varied goal-driven character-scene interactions in real-time. (2) A new method for modeling plausible action-dependent goal locations and orientations of the body given the target object geometry. (3) Incorporating explicit path planning into a variational motion synthesis network enabling navigation in cluttered scenes. (4) A new MoCap dataset with diverse human-scene interactions.

5.1 Method

Generating dynamic human scene interactions in cluttered environments requires solutions to several sub-problems. First and foremost, the synthesized motion of the

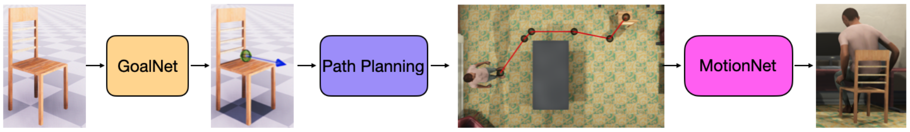


Figure 5.2: Our system consists of three main components. GoalNet predicts oriented goal locations (green sphere and blue arrow on the chair) given an interaction object. The *Path Planning Module* predicts an obstacle-free path from the starting position to the goal. *MotionNet* sequentially predicts the next character state until the desired action is executed.

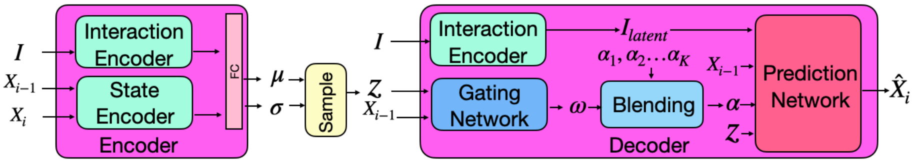


Figure 5.3: MotionNet consists of an encoder and a decoder. The encoder consists of two sub-encoders: State Encoder and Interaction Encoder. The decoder consists of a Prediction Network to predict the next character state and a gating network that predicts the blending weights of the Prediction Network. See Sec. 5.1.1.

character should be realistic and capture natural variations. Given a target object, it is important to sample plausible contact points and orientations for performing a specific action (e.g., where to sit on a chair and which direction to face). Finally, the motion needs to be synthesized such that the character navigates to the goal location while avoiding penetrating objects in the scene. Our system consists of three main components that address each of these sub-problems: a *MotionNet*, *GoalNet*, and a *Path Planning Module*. At the core of our method is the MotionNet which predicts the pose of the character based on the previous pose as well as other factors such as the interaction object geometry and the target goal position and orientation. GoalNet predicts the goal position and orientation for the interaction on the desired object. The Path Planning Module computes an obstacle-free path between the starting location of the character and the goal location. The full pipeline is illustrated in Fig. 5.2.

5.1.1 MotionNet

MotionNet is an autoregressive conditional variational autoencoder (cVAE) [43, 185] that generates the pose of the character conditioned on its previous state (e.g., pose, trajectory, goal) as well as the geometry of the interaction object. MotionNet has two components: an encoder and a decoder. The encoder encodes the previous and current states of the character and the interaction object to a latent vector Z. The decoder takes this latent vector, the character's previous state, and the interaction object to predict the character's next state. The pipeline is shown in Fig. 5.3. Note that, at test time, we only utilize the decoder of MotionNet and sample Z from a standard normal distribution.

Encoder: The encoder consists of two sub-encoders: *State Encoder* and *Interaction Encoder*. The State Encoder encodes the previous and current state of the character into a low-dimensional vector. Similarly, the Interaction Encoder encodes the object geometry into a different low-dimensional vector. Next, the two vectors are concatenated and passed through two identical fully connected layers to predict the mean μ and standard deviation σ of a Gaussian distribution representing a latent embedding space. We then sample a random latent code $\boldsymbol{Z}$, which is provided to the decoder when predicting the next state of the character.

State Representation: We use a representation similar to Starke et al. [188] to encode the state of the character. Specifically, the state at frame i is defined as $\boldsymbol{X}_i =$

$$\left\{ \boldsymbol{j}_i^p, \boldsymbol{j}_i^r, \boldsymbol{j}_i^v, \tilde{\boldsymbol{j}}_i^p, \boldsymbol{t}_i^p, \boldsymbol{t}_i^d, \tilde{\boldsymbol{t}}_i^p, \tilde{\boldsymbol{t}}_i^d, \boldsymbol{t}_i^a, \boldsymbol{g}_i^p, \boldsymbol{g}_i^d, \boldsymbol{g}_i^a, \boldsymbol{c}_i \right\}, \tag{5.1}$$

where $\boldsymbol{j}_i^p \in \mathbb{R}^{3j}, \boldsymbol{j}_i^r \in \mathbb{R}^{6j}, \boldsymbol{j}_i^v \in \mathbb{R}^{3j}$ are the position, rotation, and velocity of each joint relative to the root. j is the number of joints in the skeleton which is 22 in our data. $\tilde{\boldsymbol{j}}_i^p \in \mathbb{R}^{3j}$ are the joint positions relative to future root 1 second ahead. $\boldsymbol{t}_i^p \in \mathbb{R}^{2t}$ and $\boldsymbol{t}_i^d \in \mathbb{R}^{2t}$ are the root positions and forward directions relative to the root of frame $i-1$. $\tilde{\boldsymbol{t}}_i^p \in \mathbb{R}^{2t}$ and $\tilde{\boldsymbol{t}}_i^d \in \mathbb{R}^{2t}$ are the root positions and forward directions relative to the goal of frame $i-1$. We define these inputs for t time steps sampled uniformly in a 2 second window between $[-1, 1]$ seconds. $\boldsymbol{t}_i^a \in \mathbb{R}^{n_a t}$ is a vector of continuous action labels on each of the t samples. In our experiments, n_a is 5, which is the total number of actions we model (i.e., idle, walk, run, sit, lie down). $\boldsymbol{g}_i^p \in \mathbb{R}^{3t}, \boldsymbol{g}_i^d \in \mathbb{R}^{3t}$ are the goal positions and directions, and $\boldsymbol{g}_i^a \in \mathbb{R}^{n_a t}$ is a one-hot action label describing the action to be performed at each of the t samples. $\boldsymbol{c}_i \in \mathbb{R}^5$ are contact labels for pelvis, feet, and hands.

State Encoder: The State Encoder takes the current $\boldsymbol{X}_i$ and previous state $\boldsymbol{X}_{i-1}$ and encodes them into a low-dimensional vector using three fully connected layers.

Interaction Encoder: The Interaction Encoder takes a voxel representation of the interaction object $\boldsymbol{I}$ and encodes it into a low-dimensional vector. We use a voxel grid of size $8 \times 8 \times 8$. Each voxel stores a $4-$dimensional vector. The first three components refer to the position of the voxel center relative to the root of the character. The fourth element stores the real-valued occupancy (between 0 and 1) of the voxel. The architecture consists of three fully connected layers.

Decoder: The decoder takes the random latent code $\boldsymbol{Z}$, the interaction object representation $\boldsymbol{I}$, and the previous state $\boldsymbol{X}_{i-1}$, and predicts the next state $\hat{\boldsymbol{X}}_i$. Similar to recent work [119, 188], our decoder is built as a mixture-of-experts with two components: the Prediction Network and Gating Network.

The Prediction Network is responsible for predicting the next state $\hat{\boldsymbol{X}}_i$. The weights of the Prediction Network $\boldsymbol{\alpha}$ are computed by blending K expert weights:

$$\boldsymbol{\alpha} = \sum_{i=1}^{K} \omega_i \boldsymbol{\alpha}_i, \tag{5.2}$$

where the blending weights ω_i are predicted by the Gating Network. Each expert is a three-layer fully connected network. The Gating Network is also a three-layer fully connected network, which takes as input $\boldsymbol{Z}$ and $\boldsymbol{X}_{i-1}$.

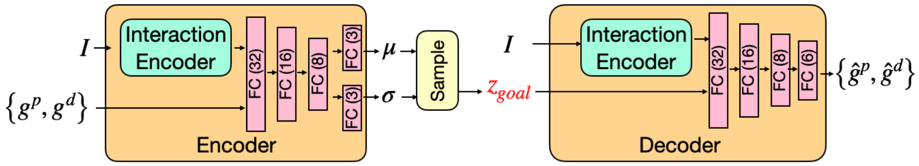


Figure 5.4: GoalNet generates multiple valid goal positions $\hat{\boldsymbol{g}}^p$ and directions $\hat{\boldsymbol{g}}^d$ given an object representation $\boldsymbol{I}$. FC(N) denotes a fully connected layer of size N.

MotionNet is trained end-to-end to minimize the loss $\mathcal{L}_{\text{motion}} =$

$$||\hat{\boldsymbol{X}}_i - \boldsymbol{X}_i||_2^2 + \beta_1 KL(Q(\boldsymbol{Z}|\boldsymbol{X}_i, \boldsymbol{X}_{i-1}, \boldsymbol{I})||p(\boldsymbol{Z})), \tag{5.3}$$

where the first term minimizes the difference between the ground truth and predicted states of the character and KL denotes the Kullback-Leibler divergence.

5.1.2 GoalNet

Given a target interaction object (which can be interactively defined by a user at test time or randomly sampled among the objects in the scene), the character is driven by the goal position $\boldsymbol{g}^p \in \mathbb{R}^3$ and direction $\boldsymbol{g}^d \in \mathbb{R}^3$ sampled on the object's surface. In order to perform realistic interactions; the character requires the ability to predict these goal positions and directions from the object geometry. For example, while a regular chair allows variation in terms of sitting direction, the direction of sitting on an armchair is restricted (see Fig. 5.11). We use GoalNet to model object-specific goal positions and directions. GoalNet is a conditional variational autoencoder (cVAE) that predicts plausible goal positions and directions given the voxel representation of the target interaction object $\boldsymbol{I}$ as shown in Fig. 5.4. The encoder encodes the interaction object $\boldsymbol{I}$, goal position $\boldsymbol{g}^p$, and direction $\boldsymbol{g}^d$, into a latent code $\boldsymbol{Z}_{goal}$. The decoder reconstructs the goal position $\hat{\boldsymbol{g}}^p$, and direction $\hat{\boldsymbol{g}}^d$ from $\boldsymbol{Z}_{goal}$ and $\boldsymbol{I}$. We represent the object using a voxel representation similar to the one used in MotionNet (Sec. 5.1.1). The only difference is that we compute the voxel position relative to the object center instead of the character root. In the encoder, we use an Interaction Encoder similar to the one used in MotionNet (see Sec. 5.1.1) to encode the object representation $\boldsymbol{I}$ to a low dimension vector. This vector is then concatenated with $\boldsymbol{g}^p$ and $\boldsymbol{g}^d$ and encoded further to the latent vector $\boldsymbol{Z}_{goal}$. The decoder has the same architecture as the encoder as shown in Fig. 5.4. The network is trained to minimize the loss:

$$\begin{aligned}\mathcal{L}_{\text{goal}} =& ||\hat{\boldsymbol{g}}^p - \boldsymbol{g}^p||_2^2 + ||\hat{\boldsymbol{g}}^d - \boldsymbol{g}^d||_2^2 \\ &+ \beta_2 KL(Q(\boldsymbol{Z}_{goal}|\boldsymbol{g}^p, \boldsymbol{g}^d, \boldsymbol{I})||p(\boldsymbol{Z}_{goal})).\end{aligned} \tag{5.4}$$

At test time, given a target object $\boldsymbol{I}$, we randomly sample $\boldsymbol{Z}_{goal} \sim \mathcal{N}(0, I)$ and use the decoder to generate various goal positions $\boldsymbol{g}^p$ and directions $\boldsymbol{g}^d$.

5.1.3 Path Planning

To ensure the character can navigate inside cluttered environments while avoiding obstacles, we employ an explicit A* path planning algorithm [69]. Given the desired goal location, we use A* to compute an obstacle-free path from the starting position of the character to the goal. The path is defined as a series of waypoints $w_i = \{w_0, w_1, w_2, ...\}$ that define the locations where the path changes direction. We break the task of performing the final desired action into sub-tasks in which each subtask requires the character to walk to the next waypoint. The final sub-task requires the character to perform the desired action at the final waypoint.

5.1.4 Training Strategy

Training MotionNet using standard supervised training produces poor quality predictions at run time. This is due to the accumulation of error at run time when the output of the network is fed back as input in the next step. To account for this, we train the network using *scheduled sampling* [15], which has been shown to result in long stable motion predictions [119]. During training, the current network prediction is used as input in the next training step with a probability $1 - P$. P is:

$$P = \begin{cases} 1 & epoch \leq C_1, \\ 1 - \frac{epoch - C_1}{C_2 - C_1} & C1 < epoch \leq C_2, \\ 0 & epoch > C2. \end{cases} \tag{5.5}$$

5.2 Data Preparation

5.2.1 Motion Data

To model variations in human-scene interactions, we capture a new dataset using an optical MoCap system with 54 Vicon cameras. We place seven different objects in the center of the MoCap area, namely two sofas, an armchair, a chair, a high bar chair, a low chair and a table. We record multiple clips of each interaction with different styles. In each sequence, the subject starts from an A-Pose in a random location in the MoCap space, walks towards the object, and performs the action for $20 - 40$ seconds. Finally, the subject gets up from the object and walks away. Our goal is to capture various styles of performing the same action, thus we ask the subject to change the style in each sequence. In addition to the subject, we also capture the object pose using attached markers. We also have the CAD model for each object. Finally, we capture running, walking, and idle sequences where the subject walks and runs in different directions with different speeds and stands in an idle state. Our dataset consists of $\sim$100 minutes of motion data recorded at 30 fps from a single subject, resulting in $\sim$185K frames. We use MoSh++ [125] to fit the SMPL-X [153] body model to the optical markers.

Fig. 5.5 shows examples of different sitting and lying down styles from our dataset. A breakdown of the dataset in terms of different actions is shown in Table 5.1. The objects used during the MoCap are shown in Fig. 5.6.

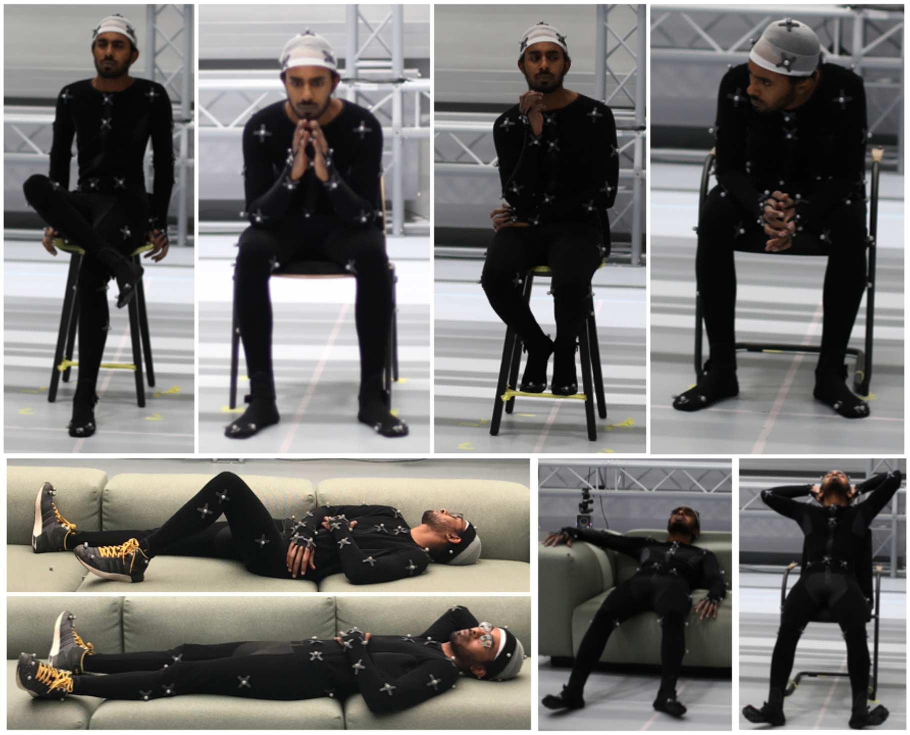


Figure 5.5: Examples of different styles in our motion capture data.

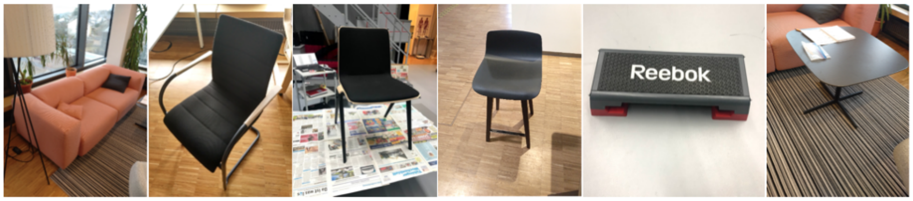


Figure 5.6: Objects used during motion capture.

Labels	Minutes	Percentage %
Idle	18.3	17.7
Walk	42.3	41.0
Run	5.1	4.9
Sit	27.3	26.4
Lie down	10.1	9.7
Total	103.3	

Table 5.1: Motion capture data breakdown with respect to actions.

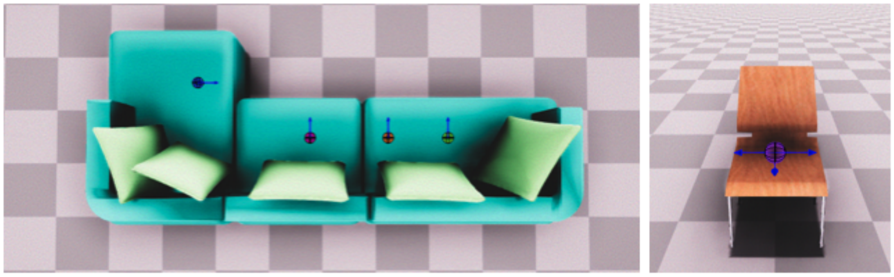


Figure 5.7: Goal Labelling.

5.2.2 Motion Data Augmentation

With only seven captured objects, MotionNet will fail to adapt to new unseen objects. Capturing MoCap with a wide range of objects requires a significant amount of effort and time. We address this issue by augmenting our data using an efficient augmentation pipeline similar to [9, 188]. Since we capture both the body motion as well as the object pose, we compute the contact between the body and the object. We detect the contacts of five key joints of the character skeleton. Namely, pelvis, hands, and feet. We then augment our data by randomly switching or scaling the object at each frame. When switching, we replace the original object with a random object of a similar size selected from ShapeNet [30]. For each new object (scaled or switched), we project the contacts detected from the ground truth data to the new object. Finally, we use an IK solver to recompute the full pose such that the contacts are maintained.

When the object is transformed, the contacts follow the same transformation. When the object is replaced by a new one, we project the original contact by finding the closest points on the surface of the new object. The new motion curve is computed by interpolation and the whole full body pose is computed using a CCD IK solver. This does not guarantee smoothness, but we found it to be stable in practice. More details are in [188].

5.2.3 Goal Data

To train GoalNet, we label various goal positions $\boldsymbol{g}^p$ and directions $\boldsymbol{g}^d$ for different objects from ShapeNet [30]. These goals represent the position on the object surface where a character could sit and the forward direction of the character when sitting. We select 5 categories from ShapeNet namely, sofas, L-shaped sofas, chairs, armchairs, and tables. From each category, we select $15-20$ instances and we manually label $1-5$ goals for each instance. Table 5.2 shows the number of instances for each category. The number of goals labeled per instance depends on how many different goals an object can afford. For example, an L-shaped sofa offers more places to sit than a chair as shown in Fig. 5.7. In total, we use 80 objects as our training data. We augment our data by randomly scaling the objects across the xyz axes leading to ∼13K training samples.

Category	Number of Objects
Armchairs	15
Chairs	16
Sofa	20
L-Sofa	18
Tables	18
Total	87

Table 5.2: GoalNet data breakdown with respect to object categories.

Network	Architecture
State Encoder	$\{512, 256, 256\}$
Interaction Encoder	$\{256, 256, 256\}$
Gating Network	$\{512, 256, 12\}$
Prediction Network	$\{512, 512, 647\}$

Table 5.3: Architecture details. All networks are all three-layer fully connected networks with ELU.

5.3 Training Details

5.3.1 MotionNet

The character state $\boldsymbol{X}$ is of size 647. The State Encoder, Interaction Encoder, Gating Network, and Prediction Network are all three-layer fully connected networks with exponential linear units (ELU). The dimensions of each network are in Table 5.3. The encoder latent code $\boldsymbol{Z}$ is of size 64 and we set the number of experts K to 12. We use a learning rate of $5e-5$ and train our network for 100 epochs. We use the Adam optimizer [102] with linear weight decay. The weight of the Kullback-Leibler divergence β_1 is 0.1.

5.3.2 GoalNet

The Interaction Encoder of GoalNet is a three-layer fully connected network of shape $\{512, 512, 64\}$. The latent vector $\boldsymbol{Z}_{goal}$ is of size 3. The weight of the Kullback-Leibler divergence β_2 is 0.5. We use the Adam optimizer with a learning rate of $1e-3$ and train GoalNet for 100 epochs.

5.3.3 Schedule Sampling

For the schedule sampling training strategy, we set $C_1 = 30$ and $C_2 = 60$. We define a roll-out window of size L where we set $L = 60$ in our experiments. For each roll-out, we feed the ground truth first frame as input to the network and then sequentially predict the subsequent frames while using the scheduled sampling strategy. We divide our training data to equal-length clips of size L.

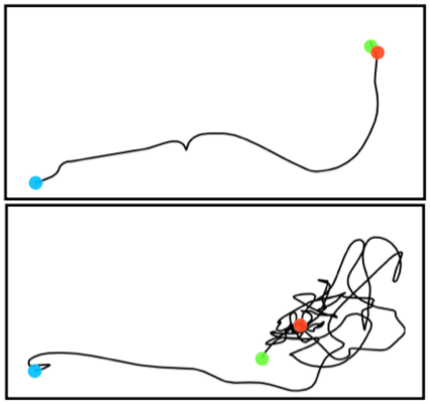


Figure 5.8: SAMP with Schedule Sampling (Top) and without (bottom). The black line shows the root projection on the xz plane. The blue and green circles denote the root at the first and last frame respectively. The red circle denotes the goal position. For both plots; the starting and goal positions are the same. Note how SAMP fails to reach the goal without the use of Schedule Sampling.

5.4 Experiments & Evaluation

5.4.1 Qualitative Evaluation

In this section, we provide qualitative results and discuss the main points.

Schedule Sampling: We found that using Schedule Sampling is essential to enable the character to successfully reach the goal and execute the action. Without it, we found the model often diverges, gets stuck, or takes a very long time to reach the goal, as we show in Fig. 5.8.

Generating Diverse Motion: In contrast to previous deterministic methods [188], SAMP generates a wide range of diverse styles of an action while ensuring realism. Several different sitting and lying down styles generated by SAMP are shown in Fig. 5.9. The use of the Interaction Encoder 5.1.1 and the data augmentation (Sec. 5.2.2) further ensures SAMP can adapt to different objects with varying geometry. Notice how the character naturally leans its head back on the sofa. The style of the action is also conditioned on the interacting object. The character lifts its legs when sitting on a high chair/table but extends its legs when sitting on a very low table. We observe that lying down is a harder task, and several of baseline methods fail to execute this task (see Sec. 5.4.2). While SAMP synthesizes reasonable sequences, our results are not always perfect. The generated motion might involve some penetration with the object.

Goal Generation: When presented with a new object, the character needs to predict where and in which direction the action should be executed. In NSM [188], the goal is computed as the object center. However, this heuristic fails for objects with complex geometries. In Fig. 5.10 we show that using the object center results in in-

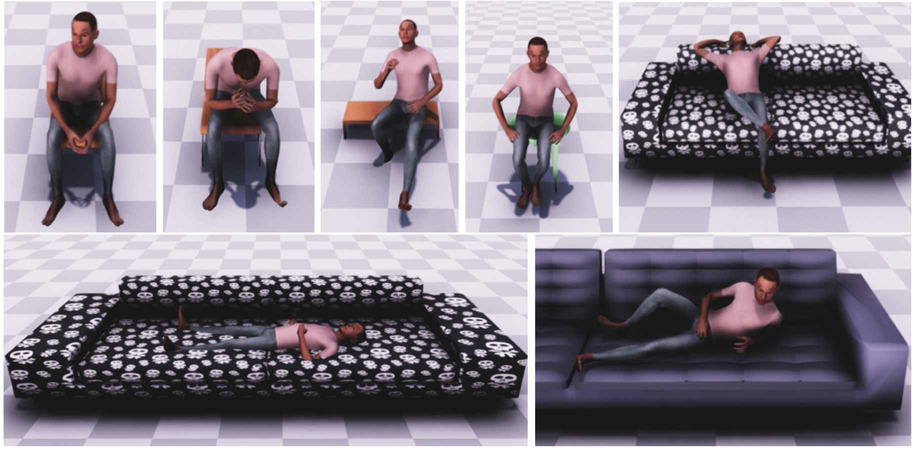


Figure 5.9: SAMP generates plausible and diverse action styles and adapts to different object geometries.

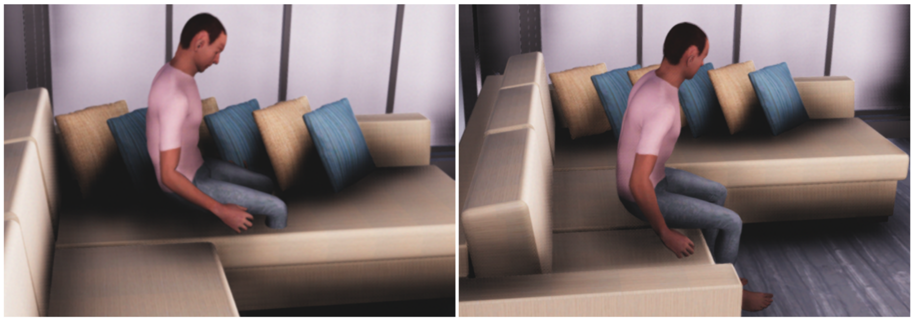


Figure 5.10: Without GoalNet (left), SAMP fails to sit on a valid place. SAMP with GoalNet is shown on the right.

valid actions, whereas GoalNet allows our method to reason about where the action should be executed. As shown in Fig. 5.11, by sampling different latent codes $\boldsymbol{Z}_{goal}$, GoalNet generates multiple goal positions and directions for various objects. Notice how GoalNet captures that, while a person can sit sideways on a regular chair, this is not valid for an armchair.

Figure 5.12 shows how the different goals generated by GoalNet guide the motion of the character. Starting from the same position, direction, and initial pose, the virtual human follows two different paths to reach different goal positions when performing the "sit on the couch" action. The final pose of the character is also different in the two cases due to the stochastic nature of MotionNet.

Path Planning: When navigating to a particular goal location in a cluttered scene, it is critical to avoid obstacles. Our Path Planning Module achieves this goal by predicting the shortest obstacle-free path between the starting character position and the

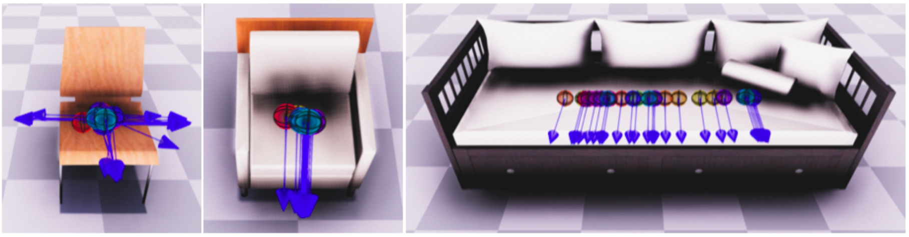


Figure 5.11: GoalNet generates diverse valid goals on different objects. Spheres indicate goal positions, and blue arrows indicate goal directions.

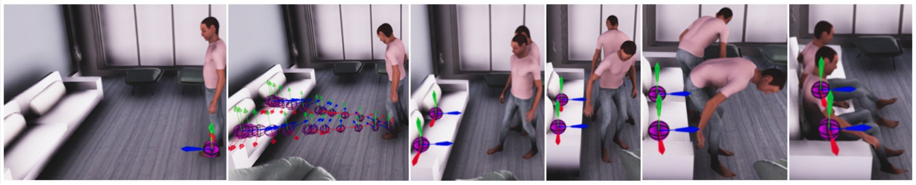


Figure 5.12: Goals generated by GoalNet (mesh spheres) are used by MotionNet to guide the motion of virtual characters.

goal. In order to use the Path Planning Module, we first compute the surface area where the character could stand or move. We call this the *navigation mesh*. This is computed from the character cylinder collider[1] and the scene geometry. The *navigation mesh*[2] is stored as convex polygons. To find a path between given start and end points, we first map these points to the closest polygons and then use A* to find the shortest path between the polygons as shown in Fig. 5.13. The navigation mesh defines the walk-able areas in the scene and is computed once offline.

In Fig. 5.14, we show an example path computed by the Path Planning Module. Without this module, the character often walks through objects in the scene. We observe a similar behavior in the previous work of NSM [188], even though NSM uses a

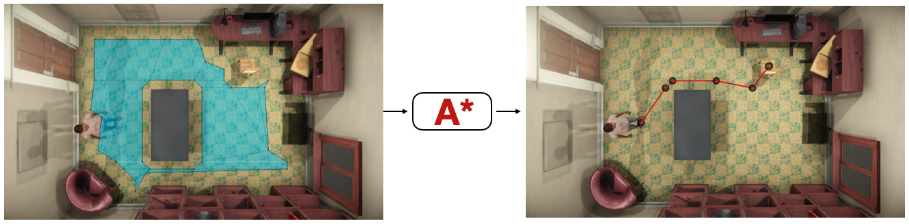


Figure 5.13: A* is used to compute an obstacle free path between the character starting position and the goal. The walk-able areas are shown in blue.

[1] https://docs.unity3d.com/Manual/class-CapsuleCollider.html

[2] https://docs.unity3d.com/Manual/nav-InnerWorkings.html

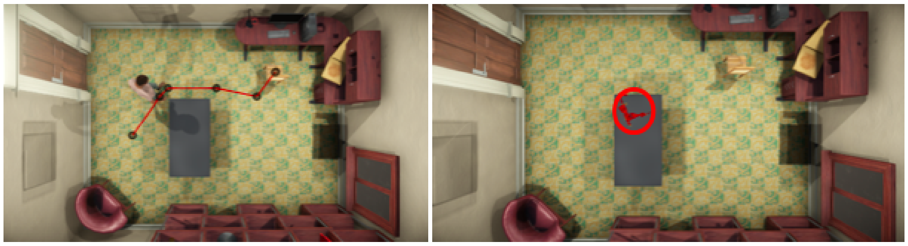


Figure 5.14: Our Path Planning Module helps SAMP to successfully navigate cluttered scenes (left). NSM [188] fails in such scenes (right).

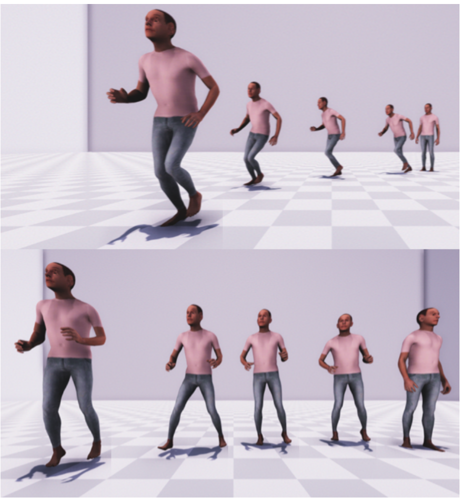


Figure 5.15: Generated running and walking motion. The user controls the motion by providing start/stop signal and the motion direction using the keyboard.

volumetric representation of the environment to help the character navigate.

Generating Controllable Motion In Fig. 5.15, we show different locomotion examples generated by SAMP given a user-specified motion direction. Notice how SAMP generates realistic head and hand motion, e.g., the character is looking in the direction of running/walking. Note that SAMP does not require phase labels [85, 188] nor a separate RL controller [119].

5.4.2 Quantitative Evaluation

Deterministic vs. Stochastic: To quantify the diversity of the generated motion, we put the character in a fixed starting position and direction and we run our method ten times with the same goal. For example, we instruct the character to sit/lie down on the same object multiple times starting from the same initial state/position/direction. For walking and running, we instruct the character to run in each of the four directions for

	Walk	Run	Sit	Liedown
Ground-truth	5.95	7.74	5.18	7.52
SAMP	5.63	5.75	5.05	6.69

Table 5.4: Diversity metric. Higher values indicate more diversity.

15 seconds. We record the character motion for each run and then compute the Average Pairwise Distance (APD) [221, 229] as shown in Table 5.4. The APD is defined as:

$$APD = \frac{1}{N(N-1)} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} ||\boldsymbol{X}'_i - \boldsymbol{X}'_j||_2^2. \tag{5.6}$$

where $\boldsymbol{X}'_i$ represents the character's local pose features at frame i. $\boldsymbol{X}'_i = \{\boldsymbol{j}_i^p, \boldsymbol{j}_i^r, \boldsymbol{j}_i^v\}$. N is the total number of frames for all sequences. For comparison, we also report the APD for the ground-truth (GT) data in Table 5.4.

GoalNet: Given 150 unseen goals sampled on test objects, we measure the average position and orientation reconstruction error of GoalNet to be $\mathbf{6.04}$ cm and $\mathbf{2.29}$ deg (we note that the objects have real-life measurements). To measure the diversity of the generated goals, we compute the Average Pairwise Distance (APD) among the generated goal positions g^p and directions g^d:

$$\text{APD-Pos} = \frac{1}{LN(N-1)} \sum_{k=0}^{L} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} |g_i^p - g_j^p| \tag{5.7}$$

$$\text{APD-Rot} = \frac{1}{LN(N-1)} \sum_{k=0}^{L} \sum_{i=0}^{N} \sum_{\substack{j=0 \\ j \neq i}}^{N} \arccos(g_i^d . g_j^d). \tag{5.8}$$

$L = 150$ is the number of objects and $N = 10$ is the number of goals generated for each object. We find APD-Pos and APD-Rot for our generated goals to be $\mathbf{16.42}$ cm and $\mathbf{41.27}$ deg compared to $\mathbf{16.18}$ cm and $\mathbf{90.23}$ deg for the ground-truth (GT) data.

Path Planning Module: To quantitatively evaluate the effectiveness of our Path Planning Module, we test our method in a cluttered scene. We put the character in a random initial position and orientation and select a random goal. We repeat this 10 times. We find the percentage of frames where a penetration happens is $\mathbf{3.8}\%$, $\mathbf{11.2}\%$, and $\mathbf{8.11}\%$ for SAMP with Path Planning Module, without Path Planning Module, and NSM [188], respectively. While NSM uses a volumetric sensor to detect collisions with the environment, it is not as effective as explicit path planning.

Interaction Encoder Ablation: To quantify the importance of the Interaction Encoder, we train SAMP without the Interaction Encoder. We find that the precision of reaching the goal deteriorates to 14.82 cm and 3.65 deg compared to 6.09 cm and 3.55 deg when the Interaction Encoder is used.

Comparison to Previous Models: We compare our model to baselines by measuring three metrics: average execution time, average precision, and Frèchet distance

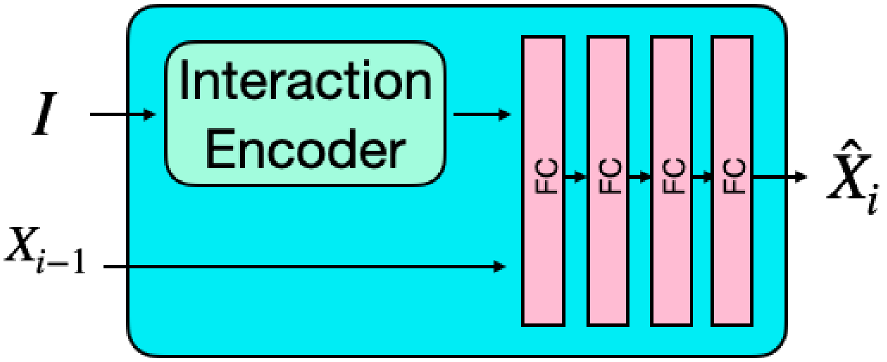


Figure 5.16: MLP Architecture.

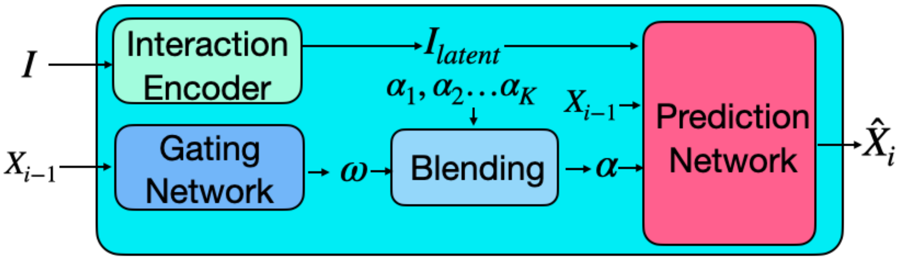


Figure 5.17: MoE Architecture.

(FD) [44] between the distribution of the generated motion and ground-truth. Execution time is the time required to transition to the target action label from an idle state. Precision is the positional (PE) and rotational (RE) error at the goal. We measure FD on a subset of the state features which we call $\tilde{\boldsymbol{X}}$:

$$\tilde{\boldsymbol{X}} = \{\boldsymbol{j}^p, \boldsymbol{j}^r, \boldsymbol{j}^v, \tilde{\boldsymbol{t}}^p, \tilde{\boldsymbol{t}}^d\}. \tag{5.9}$$

As our baselines, we choose a feed-forward network (MLP) as the motion prediction network, Mixture of Experts (MoE) [225], and NSM [188]. The architecture of the MLP is shown in Fig. 5.16. We use the same Interaction Encoder used for our MotionNet followed by four fully connected layers of size 512. The architecture of the MoE is shown in Fig. 5.17. The Interaction Encoder, Gating Network, and Prediction Network are all the same as the one used in MotionNet.

SAMP vs. MLP vs. MoE: We re-trained the MLP and MoE using the same training strategy and data we used for SAMP. Both MLP and MoE take a longer time to execute the task and often fail to execute the "lie down" action (denoted ∞) as evidenced by the execution time in Table 5.5 and precision in Table 5.6. These architectures sometimes generate implausible poses, which is reflected by the lower FD in Table 5.7

	Sit	Liedown
MLP	13.06	∞
MoE	12.99	∞
SAMP	**12.53**	**17.06**
Ground-truth	11.70	15.49

Table 5.5: Average execution Time in seconds. ∞ means the method failed to reach the goal within 3 minutes.

Method	Sit		Liedown	
	PE(cm)	RE(deg)	PE(cm)	RE(deg)
MLP	9.27	3.99	∞	∞
MoE	7.99	5.73	∞	∞
SAMP	**6.09**	**3.55**	**5.76**	**6.45**

Table 5.6: Average precision in terms of positional and rotational errors (PE and RE). ∞ means the method failed to reach the goal within 3 minutes.

	Idle	Walk	Run	Sit	Liedown
MLP	102.85	121.18	150.56	105.87	36.85
MoE	102.91	114.17	151.14	105.10	35.79
SAMP	**102.72**	**111.09**	**141.11**	**104.68**	**17.30**

Table 5.7: Frèchet distance.

Metric	Sit		Carry	
	SAMP	NSM	SAMP	NSM
Precision PE (cm) ↓	**15.97**	16.95	**4.58**	4.72
Precision RE (deg) ↓	5.38	**2.32**	1.78	**1.65**
Execution Time (sec) ↓	12.93	**10.26**	13.29	**12.82**
FD ↓	6.20	**4.21**	10.17	**7.31**
Diversity ↑	**0.44**	0.0	**0.26**	0.0
Penetration (%) ↓	**3.8**	8.11	**3.62**	8.45

Table 5.8: SAMP vs. NSM.

SAMP vs. NSM: For NSM, we used the publicly available pre-trained model since retraining NSM on our data is infeasible due to the missing phase labels. We trained SAMP on the same data on which NSM was trained. In Table 5.8 we observe that our model is on par with NSM in terms of achieving goals without the need for phase labels, which are cumbersome and often ambiguous to annotate. In addition, our main focus is to model diverse motions via a stochastic model while NSM is deterministic. Our Path Planning Module module helps SAMP to safely navigate complex scenes where NSM fails as shown by the penetration amounts.

For all evaluations, all test objects are randomly selected from ShapeNet and none is part of our training set.

Comparison to Cao et al.: While relevant, the formulation of Cao [25] et al. is significantly different from our method, making a direct comparison difficult. Given a target interaction object and action (e.g. "sit on the couch"), SAMP samples a goal location and orientation on the object, computes an obstacle-free path towards the object, and synthesizes diverse motion sequences that are of arbitrary length until the goal is executed. We assume that the character starts the action from an idle position without any knowledge of the past. In contrast, Cao et al. sample a goal *location* in the image space given a one-second-long *history* of motion. Based on this trajectory, a deterministic motion sequence of fixed length (two-seconds) is synthesized. The action executed in this trajectory is not controllable.

5.5 Limitations and Future Work

We observe that sometimes slight penetrations between the character and the interacting object can occur. A potential solution is to incorporate a post-processing step to optimize the pose of the character to avoid such intersections. SAMP might not adapt well to objects with significantly different geometry than those seen in training as shown in Fig. 5.18. In order to generalize SAMP to interacting objects that have significantly different geometry than those seen in training, future work should explore methods to encode local object geometries.

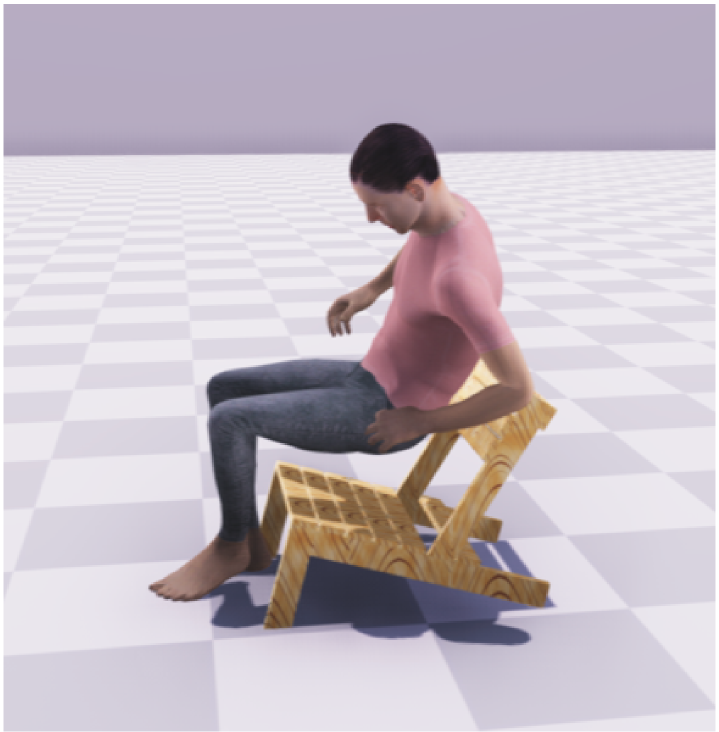


Figure 5.18: SAMP with significantly different geometry.

5.6 Conclusion

Here we have described SAMP, which makes several important steps toward creating lifelike avatars that move and act like real people in previously unseen and complex environments. Critically, we introduce three elements that must be part of a solution. First, characters must be able to navigate the world and avoid obstacles. For this, we use an existing path planning method. Second, characters can interact with objects in different ways. To address this, we train GoalNet to take an object and stochastically produce an interaction location and direction. Third, the character should produce motions achieving the goal that vary naturally. To that end, we train a novel MotionNet that incrementally generates body poses based on the past motion and the goal. We train SAMP using a novel dataset of motion capture data involving human-object interaction.

Chapter 6

Synthesizing Physical Character-Scene Interactions

Existing techniques for synthesizing character-scene interactions tend to be limited in terms of motion quality, generalization, or scalability. Traditional motion blending and editing techniques [57, 110] require significant manual effort to adapt existing motion clips to a new scene. Data-driven kinematic models, e.g. SAMP (Chapter 5), can produce high-quality motion when applied in environments similar to those shown in the dataset. However, when applied to new scenarios, kinematic models can struggle to generate realistic behaviors that respect scene constraints. Furthermore, kinematic methods [74, 85, 188, 225] are also prone to producing motion artifacts, such as floating, sliding, or penetrating other objects in the scene. Physics-based methods are able to better synthesize plausible motions in new scenarios by leveraging a physics simulation of a character's movements and interactions within a scene. Reinforcement learning (RL) has become one of the most commonly used paradigms for developing control policies for physically simulated characters. However, it can be notoriously difficult to design RL objectives that lead to high-quality and natural motions [80]. Motion tracking [155] can improve motion quality by training control policies to imitate reference motion data. However, it can be difficult to apply tracking-based methods to complex scene-interaction tasks, where a character may need to compose and transition between a diverse set of skills in order to effectively interact with its surroundings.

Recently, Adversarial Motion Priors (AMP) [157] have been proposed as means of imitating behaviors from large unstructured motion datasets, without requiring any annotation of the motion data or an explicit motion planner. This method leverages an adversarial discriminator to differentiate between motions in the dataset and motions generated by the policy. The policy is trained to satisfy a task reward while also trying to fool the discriminator by producing motions that resemble those shown in the dataset. Crucially, the policy need not explicitly track any particular motion clip, but is instead trained to produce motions that are within the distribution of the dataset. This allows the policy to deviate, interpolate, and transition between different behaviors as needed to adapt to new scenarios. This versatility is crucial for character-scene interac-

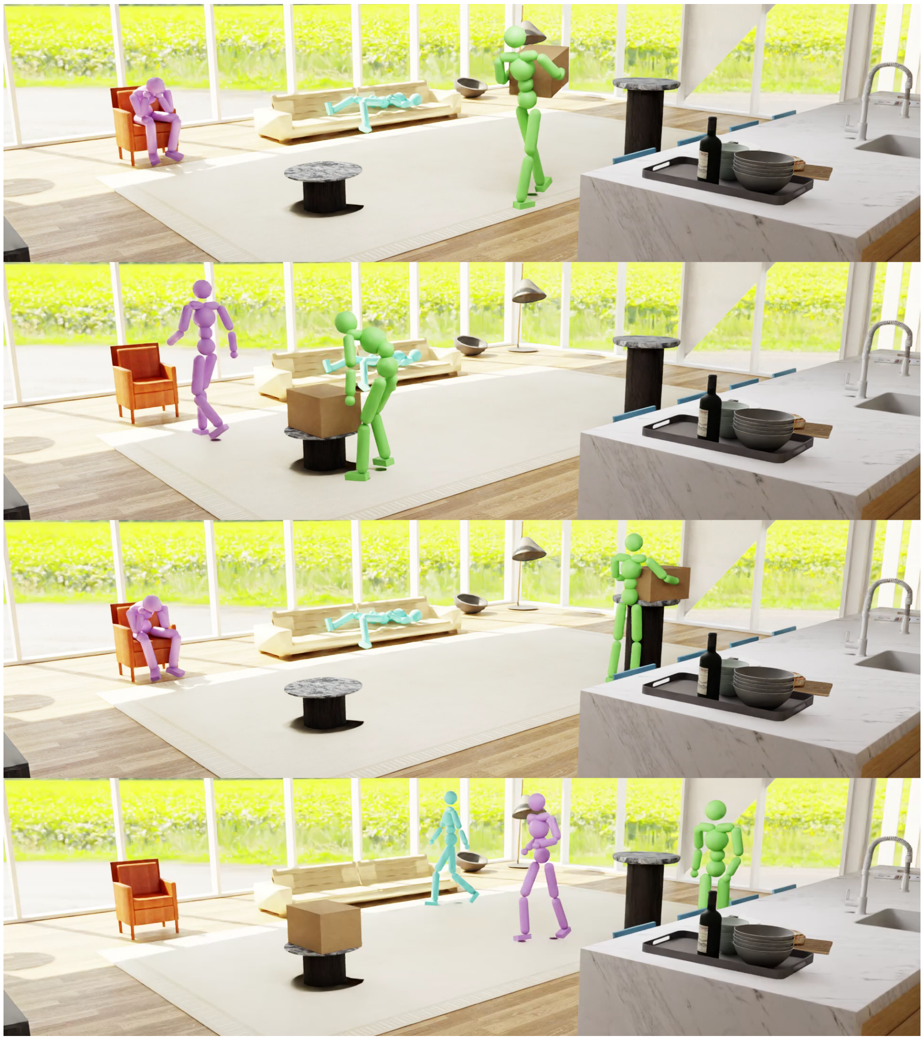


Figure 6.1: InterPhys enables physically simulated characters to perform scene interaction tasks in a natural and life-like manner. We demonstrate the effectiveness of our approach through three challenging scene interaction tasks: carrying, sitting, and lying down, which require coordination of a character's movements in relation to objects in the environment.

tion, which requires fine-grain adjustments to a character's behaviors in order to adapt different object configurations within a scene.

In this work, we present a framework for training physically simulated characters to perform scene interaction tasks. Our method builds on AMP and extends it to character-scene interaction tasks. Unlike the AMP discriminator, which only considers the character's motion, our discriminator jointly examines the character and the object in the scene. This allows our discriminator to evaluate the realism of the character's movements within the context of a scene (e.g., a sitting motion is realistic only when a chair is present). In addition, given a small dataset of human-scene interactions, our policy discovers how to adapt these behaviors to new scenes. For example, from about five minutes of motion capture data of a human carrying a single box, we are able to train a policy to carry hundreds of boxes with different sizes and weights. Similarly, from a few demonstrations of lying down on a single sofa, our policy discovers how to lie down on several different sofas and beds. We achieve this by populating our simulated environments with a wide range of object instances and randomizing their configuration and physical properties. By interacting with these rich simulated environments, our policies learn how to realistically interact with a wide range of object instances and environment configurations. We demonstrate the effectiveness of our method with three challenging scene-interaction tasks: sit, lie down, and carry. Examples of the three tasks are shown in Fig. 6.1. As we show in our experiments, our policies are able to effectively perform all of these tasks and achieve superior performance compared to prior state-of-the-art kinematic-based and physics-based methods.

6.1 Method

To train policies that enable avatars to interact with objects in a natural and life-like manner, we build on the Adversarial Motion Priors (AMP) framework [157]. Our approach consists of two components: a policy and a discriminator as shown in Fig. 6.2. The discriminator's role is to differentiate between the behaviors produced by the simulated character and the behaviors depicted in a motion dataset. The role of the policy π is to control the movements of the character in order to maximize the expected accumulative reward $J(\pi)$. The agent's reward r_t at each time step t is specified according to:

$$r_t = w^G r^G(\mathbf{s}_t, \mathbf{a}_t, \mathbf{g}_t) + w^S r^S(\mathbf{s}_t, \mathbf{s}_{t+1}). \tag{6.1}$$

The task reward r^G encourages the character to satisfy high-level objectives, such as sitting on a chair or moving a box to the desired location. The style reward r^S encourages the character to imitate behaviors from a motion dataset as it performs the desired task. $\mathbf{s}_t \in \mathcal{S}$ is the state at time t. $\mathbf{a}_t \in \mathcal{A}$ are the actions sampled from the policy π at time step t. $\mathbf{g}_t \in \mathcal{G}$ denotes the task-specific goal features at time t. w^G and w^S are empirical weights.

The policy is trained to maximize the expected discount return $J(\pi)$,

$$J(\pi) = \mathbb{E}_{p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \tag{6.2}$$

where $p(\tau|\pi)$ denotes the likelihood of a trajectory τ under the policy π. T is time horizon, and $\gamma \in [0, 1]$ is a discount factor.

The discriminator measures the similarity between the motions produced by the physically simulated character and the motions depicted in a dataset of motion clips.

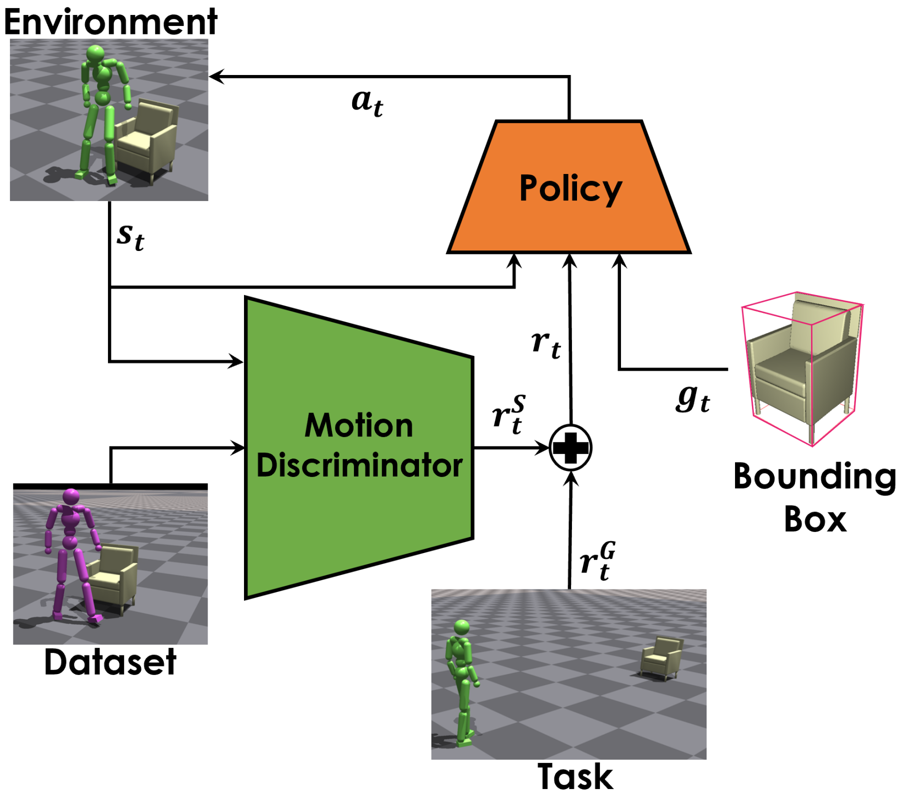


Figure 6.2: InterPhys has two main components: a policy and a discriminator. The discriminator differentiates between the behaviors generated by the policy and the behaviors depicted in a motion dataset. In contrast to prior work, the discriminator and policy take information about the scene object into account. Specifically, the policy takes the previous character and object states, $\mathbf{s}_t$, in addition to the object bounding box, and controls character movements to achieve a task reward r^G while producing a motion that looks realistic to the discriminator.

The discriminator is trained according to the objective proposed by Peng et al. [157]:

$$\arg\min_{D} \; -\mathbb{E}_{d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})}\left[\log\left(D(\mathbf{s},\mathbf{s}_{t+1})\right)\right] \tag{6.3}$$

$$-\mathbb{E}_{d^{\pi}(\mathbf{s},\mathbf{s}_{t+1})}\left[\log\left(1-D(\mathbf{s},\mathbf{s}_{t+1})\right)\right] \tag{6.4}$$

$$+\, w^{\mathrm{gp}}\, \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})}\left[\left|\left|\nabla_{\phi} D(\phi)\big|_{\phi=(\mathbf{s},\mathbf{s}_{t+1})}\right|\right|^2\right], \tag{6.5}$$

where $d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})$ and $d^{\pi}(\mathbf{s},\mathbf{s}_{t+1})$ represent the likelihoods of the state transition from $\mathbf{s}$ to $\mathbf{s}_{t+1}$ under the dataset distribution $\mathcal{M}$ and the policy π respectively. w^{gp} is a manually specified coefficient for a gradient penalty regularizer [133]. The style reward r^S for the policy is then specified according to:

$$r^S(\mathbf{s}_t,\mathbf{s}_{t+1}) = -\log(1-D(\mathbf{s}_t,\mathbf{s}_{t+1})). \tag{6.6}$$

6.2 State and Action Representation

The state $\mathbf{s}$ is represented by a set of features that describe the configuration of the character's body, as well as the configuration of the objects in the scene relative to the character. These features include:

- Root height
- Root rotation
- Root linear and angular velocity
- Local joints rotations
- Local joints velocities
- Positions of four key joints: right hand, left hand, right foot, and left foot
- Object position
- Object orientation

The height and rotation of the root are recorded in the world coordinate frame while velocities of the root are recorded in the character's local coordinate frame. Rotations are presented using a 6D normal-tangent encoding [157]. The positions of four key joints, object position, and object orientation are recorded in the character's local coordinate frame. A key difference from prior work is the inclusion of object features in the state. These object features enable the discriminator to not only judge the realism of the motion, but also how realistic the motion is w.r.t. to the object. Note that the object can move during the action and the agent must react appropriately. Combined, these features result in a 114D state space. The actions $\mathbf{a}$ generated by the policy specify joint target rotations for PD controllers. Each target is represented as an exponential map $\mathbf{a} \in \mathbb{R}^3$ [61], resulting in a 28D action space.

6.3 Tasks

We aim to train simulated character to solve character-scene interaction tasks. To demonstrate the effectiveness of our method; we choose three challenging interactive tasks: sit, lie down, and carry. The style reward r^S is the same for all tasks as defined in Eq. 6.6. The task reward r^G is task-specific as detailed in the following subsections.

6.3.1 Sit

The objective of this task is for the character to move to a target object and to sit on it. The object is initialized at a random orientation anywhere between one and ten meters away from the character.

The goal $\mathbf{g}_t \in \mathbb{R}^3$ is the object bounding box. The task reward is defined as :

$$r_t^G = \begin{cases} 0.7\, r_t^{\text{near}} + 0.3\, r_t^{\text{far}}, & ||\mathbf{x}^* - \mathbf{x}_t^{\text{root}}|| > 0.5m \\ 0.7\, r_t^{\text{near}} + 0.3, & \text{otherwise} \end{cases} \tag{6.7}$$

where $\mathbf{x}^{\text{root}}$ is the position of the character's root, $\mathbf{x}^*$ is the object position, r^{far} encourages the character to walk towards the object, while r^{near} encourages the character to sit on the object once it is close by. r^{far} is specified according to:

$$\begin{aligned} r_t^{\text{far}} &= 0.5 \exp\left(-0.5||\mathbf{x}^* - \mathbf{x}_t^{\text{root}}||^2\right) \\ &\quad + 0.5 \exp\left(-2.0||\, v^* - \mathbf{d}_t^* \cdot \dot{\mathbf{x}}_t^{\text{root}}||^2\right) \end{aligned} \tag{6.8}$$

where $\dot{\mathbf{x}}_t^{\text{root}}$ is the linear velocity of the character's root, $\mathbf{d}^*$ is a horizontal unit vector pointing from the root $\mathbf{x}_t^{\text{root}}$ to the object's location $\mathbf{x}^*$, and $v^* = 1.5m/s$ is the target speed at which the character should walk. Once the character is close to the object, r^{near} is used to encourage the character to sit on the object:

$$r_t^{\text{near}} = \exp\left(-10.0||\mathbf{x}^{\text{root}*} - \mathbf{x}_t^{\text{root}}||^2\right), \tag{6.9}$$

with $\mathbf{x}^{\text{root}*}$ denoting the target sitting position on the object where the character's hip should be placed.

6.3.2 Lie down

The objective of the lie down task is for the character to walk towards an object and then lie down on it. The goal $\mathbf{g}_t$ and the task reward r_t^G are the same as for the sitting task (see Eq. 6.7). r^{far} is defined as in Eq. 6.8, and $r_t^{\text{near}} =$

$$\exp\left(-10.0||\mathbf{x}^{\text{root}*} - \mathbf{x}_t^{\text{root}}||^2 - 10.0||h^{\text{head}*} - h_t^{\text{head}}||^2\right) \tag{6.10}$$

where h^{head} is the height of the character's head, and $h^{\text{head}*}$ is the target head height.

6.3.3 Carry

The objective of the carry task is for the character to pick up a box and carry it to a destination. The goal is specified according to:

$$\mathbf{g}_t = (\tilde{\mathbf{x}}'_t, b_h, b_w, b_d), \tag{6.11}$$

where $\tilde{\mathbf{x}}'_t$ is the target position on which the box should be placed; $\tilde{\mathbf{x}}'_t$ is represented in the character's local coordinate frame. b_h, b_w, b_d are the box height, width, and depth respectively. The task reward is specified according to:

$$r^G_t = r^{\text{walk}}_t + r^{\text{carry}}_t, \tag{6.12}$$

where r^{walk} encourages the character to walk towards the box and stay close to it. More specifically, it encourages the character to move its root $\mathbf{x}^{\text{root}}_t$ towards the position of the box $\mathbf{x}^*_t$ at a target speed $\mathbf{d}^*$:

$$r^{\text{walk}}_t = \begin{cases} 0.1 \exp\left(-0.5||\mathbf{x}^*_t - \mathbf{x}^{\text{root}}_t||^2\right) + \\ 0.1 \exp\left(-2.0||\ v^* - \mathbf{d}^*_t \cdot \dot{\mathbf{x}}^{\text{root}}_t||^2\right), & ||\mathbf{x}^*_t - \mathbf{x}^{\text{root}}_t|| > 0.5m \\ 0.2, & \text{otherwise} \end{cases} .$$

r^{carry} encourages the character to carry the box to a target position $\mathbf{x}'_t$:

$$r^{\text{carry}}_t = \begin{cases} r^{\text{carry}-\text{far}}_t + r^{\text{carry}-\text{near}}_t, & ||\mathbf{x}'_t - \mathbf{x}^*_t|| > 0.5m \\ 0.2 + r^{\text{carry}-\text{near}}_t, & \text{otherwise} \end{cases} . \tag{6.13}$$

$r^{\text{carry}-\text{far}}_t$ is defined as:

$$\begin{aligned} r^{\text{carry}-\text{far}}_t &= 0.2 \exp\left(-0.5||\mathbf{x}'_t - \mathbf{x}^*_t||^2\right) \\ &+ 0.2 \exp\left(-2.0||\ v' - \mathbf{d}'_t \cdot \dot{\mathbf{x}}^{\text{box}}_t||^2\right) \\ &+ 0.1 \exp\left(-10.0||h^{\text{hand}}_t - h^{\text{box}}_t||^2\right). \end{aligned} \tag{6.14}$$

Where $\mathbf{d}'_t$ is a horizontal unit vector pointing from the box location to the target location, $\dot{\mathbf{x}}^{\text{box}}_t$ is the velocity of the box, $v' = 1.5m/s$ is the target speed. h^{hand}_t and h^{box}_t are the height of the character's hand and box height respectively. Once the box is close to the target, $r^{\text{carry}-\text{near}}_t$ encourages the character to place the box precisely on the target platform,

$$r^{\text{carry}-\text{near}}_t = 0.2 \exp\left(-10.0||\mathbf{x}'_t - \mathbf{x}^*_t||^2\right). \tag{6.15}$$

6.4 Data

To train the character to interact with objects in a life-like manner, we train our motion priors using demonstration data of human-scene interactions. For the sit and lie down tasks; we use the SAMP dataset [74], which contains 100 minutes of MoCap clips of sitting and lying down behaviors. In addition to the human motion, the SAMP dataset also records the positions and orientations of the objects in the scene. The dataset also provides CAD models for seven different objects. For the carry task; we captured 15 MoCap clips of a subject carrying a single box. In each clip, the subject walks towards the box, picks it up, and carries it to a target location. The initial and target box locations are varied in each clip. In addition to full-body MoCap, we track the box motion using optical markers on the box.

The SAMP dataset provides examples of interactions with only seven objects, similarly our object-carry dataset only contains demonstration of carrying a single box. Nonetheless, we show that our reinforcement learning framework allows the agent to generalize from these limited demonstrations to interact with a much wider array of objects in a natural manner. This is achieved by exposing the policy to new objects in the training phase. Our policy is trained using multiple environments simulated in parallel in IsaacGym [127]. We populate each environment with a different object instance to encourage our policy to learn how to interact with objects exhibiting natural class variation. For the sit and lie down tasks, we replace the original objects with different objects of the same class from ShapeNet [30]. The categories are: regular chairs, armchairs, tables, low stools, high stools, sofas, and beds. In total, we used ~ 350 unique objects from ShapeNet [30]. We increase the variability between the objects even further by randomly scaling the objects in each training episode with a scale factor between 0.8 and 1.2. For the carry task; the size of the object is randomly scaled for each environment. The scale is sampled uniformly between 0.5 and 1.5.

6.5 Training

At the start of each episode, the character and object are initialized to states sampled randomly from the dataset. Both states are sampled with the same timestamp. This leads to the character sometimes being initialized in a standing state, requiring it to learn to walk towards the target and execute the action. At other times, it is initialized close to the final state and just has to learn to maintain its state, i.e. sitting on the object or holding a box. In contrast to always initializing the policy to a fixed starting state (e.g. standing), this Reference State Initialization (RSI) [155] has been shown to significantly speed up training progress and produce more realistic motions. The reason is that the policy gets to see the desired final state early on.

Since the reference motions are limited; initialization from these alone is not sufficient to cover all possible starting positions or orientations. We would like our policy to be able to execute the desired task from a wide range of positions and orientations. We achieve this by randomizing the object position w.r.t. the character at the beginning of each episode. The object is placed anywhere between one and ten meters away from the character on the xy plane. The object orientation is sampled uniformly between 0 and 2π.

The episode length is set to 10 seconds for the sit and lie down tasks, and it is 15 seconds for the carry task. The episode is terminated once its duration is exceeded. In addition, we terminate the policy early if any joint, except the feet and hands, is within 20 cm of the ground, or if the box is within 30 cm of the ground.

The policy π is a neural network that takes the current state $\mathbf{s}_t$ and goal $\mathbf{g}_t$

and predicts the mean $\mu(\mathbf{s}_t, \mathbf{g}_t)$ of a Gaussian action distribution $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g}_t) = \mathcal{N}(\mu(\mathbf{s}_t, \mathbf{g}_t), \Sigma)$. The covariance matrix Σ is set manually and kept fixed during training. The policy, value function and the discriminator are all three fully connected networks with the following dimensions $\{1024, 512, 28\}$, $\{1024, 512, 1\}$, $\{1024, 512, 1\}$ respectively. ReLu is used between all hidden layers in the three networks. We follow the training strategy of Peng et al. [157] to jointly train the policy and the discriminator. Our policy is trained using proximal-policy optimization (PPO) [178].

6.6 Results

In this section, we show results of our method on different scene-interaction tasks. In Fig. 6.3 we show examples of our character executing sit, lie down, and carry tasks. In each task the character is initialized far from the object with a random orientation. First, the character approaches the object, using locomotion skills like walking and running, and then seamlessly transitions to task-specific behavior, such as sitting, lying down, or picking up the object. The character is able to smoothly transition from idling to walking, and from walking to the various task-specific behaviors.

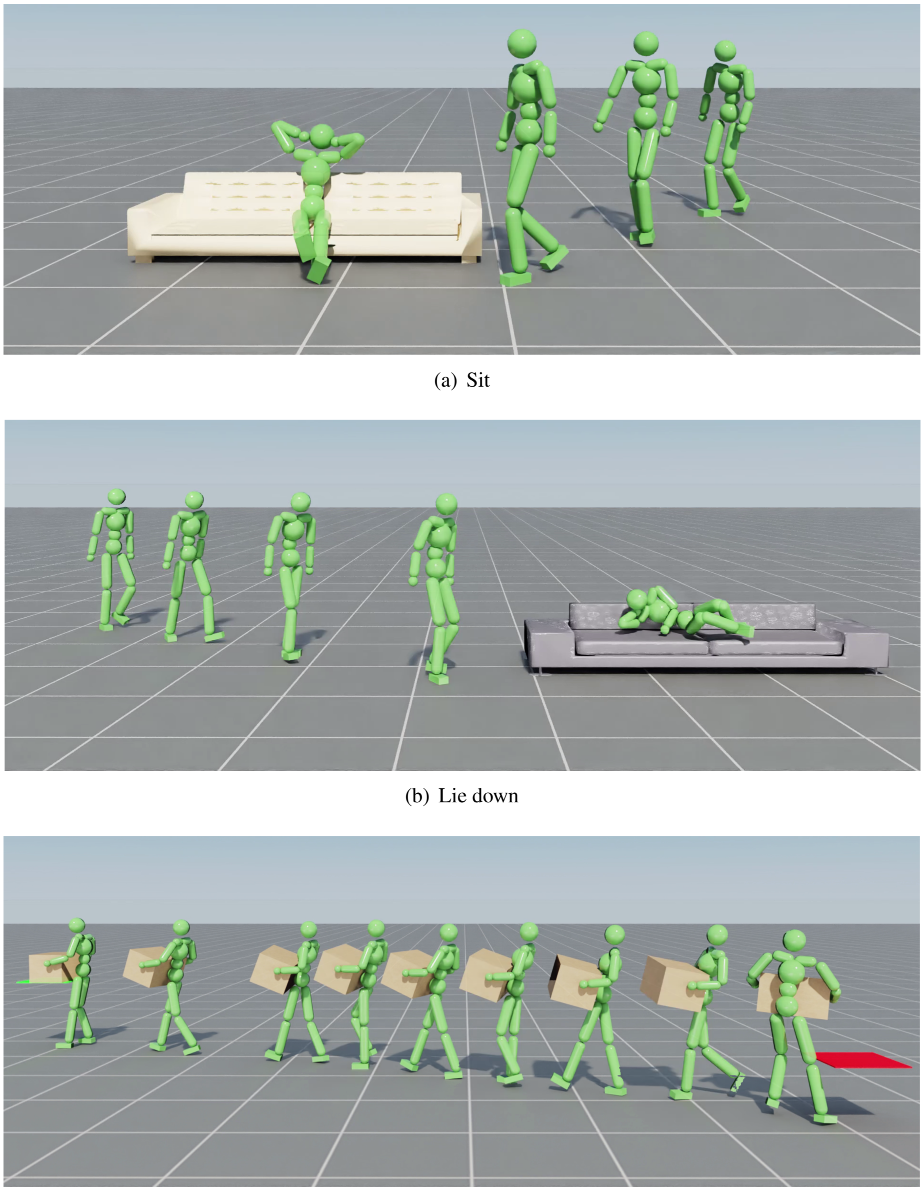
From human demonstrations of interacting with eight objects only, we teach our policy to sit and lie down on ~ 350 training objects. We demonstrate the generalization power of our model by testing on objects that were not seen during training as shown in Fig. 6.4. Our method successfully sits and lies down on a wide range of objects and is able to adapt the character's behaviors to a given object. The character jumps to sit on a high chair, leans back on a sofa, and puts its arms on the armrests of a chair when present. We used ~ 350 training objects and tested on 21 objects. Similarly, our policy learns to carry boxes of different sizes as shown in Fig. 6.5. We tested our policy on box sizes sampled uniformly between $25 \times 17.5 \times 15$ cm and $75 \times 52.5 \times 45$ cm. Our method generalizes beyond what is shown in the human demonstrations. For example, the character can carry very small boxes as shown in Fig. 6.5 although no such objects were depicted in the human demonstration dataset.

We further test our policy on different scales of the same object as shown in Fig. 6.6. We observe that the policy learns to seamlessly adapt to the different object sizes, and it succeeds in placing its hip on the proper support surface.

Next, we study how our policy deals with objects of different physical properties. We train a policy to carry boxes of the same size but different weights. The weights are sampled uniformly between 5 kg and 26 kg. For this experiment, we augment the goal $\mathbf{g}_t$ with the box density. Some examples are shown in Fig. 6.7 where heavier boxes are indicated with darker colors. The character discovers how to deal with the different weights from a human demonstration of a single box.

Humans have the ability to interact with the same object in a myriad of different styles. As shown in Fig. 6.8, our character also demonstrates similar diversity in its interactions with a given object. The character exhibits different styles while sitting, including regular sitting, leaning backwards, or sitting with different arms movements.

We quantitatively evaluate our method by measuring the success rate for each task. Table 6.1 summarizes the performance statistics on the various tasks. Success rate records the percentage of trails where the character successfully completes the task



(a) Sit

(b) Lie down

(c) Carry

Figure 6.3: Our method successfully executes three challenging scene-interaction tasks in a life-like manner.

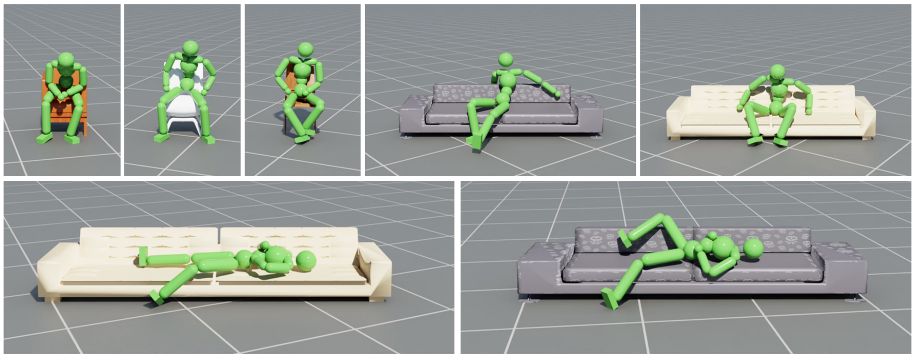


Figure 6.4: Our method successfully sits and lies down on a wide range of objects and is able to adapt the character's behaviors to a given object.

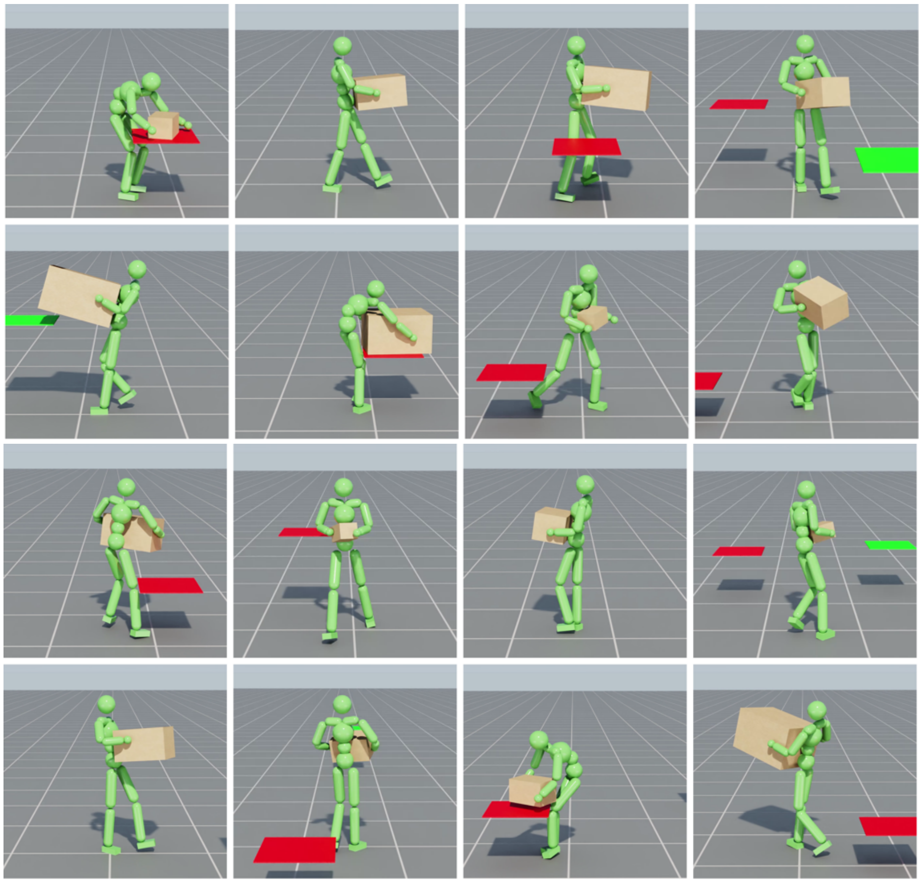


Figure 6.5: Our method generalizes beyond the human demonstrations and learns to carry different boxes from a human demonstration of carrying a single box.

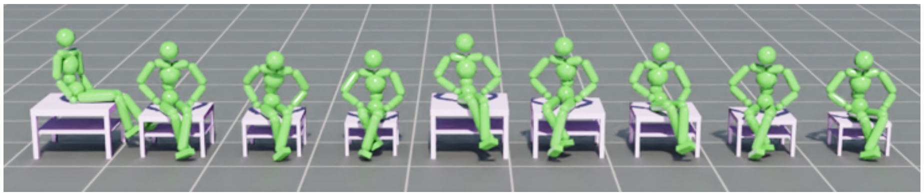


Figure 6.6: Our policy seamlessly adapt to the different object sizes.

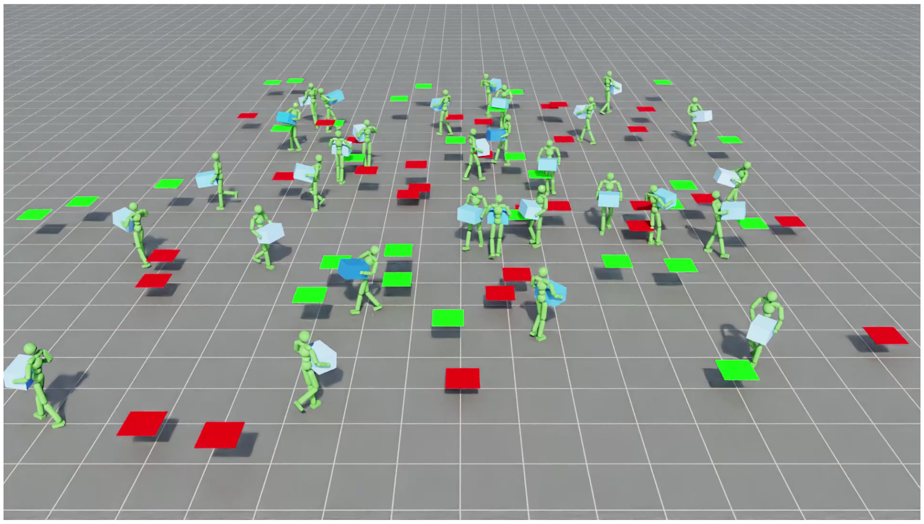


Figure 6.7: Carrying boxes of different weights. Darker colors indicate heavier weights.

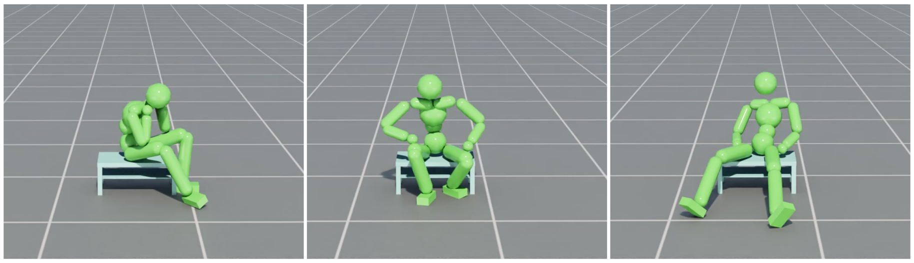


Figure 6.8: Different styles on the same object.

Task	**Success Rate (%)**	**Execution Time (Seconds)**	**Precision (cm)**
Sit	90.4	5.0	6.7
Lie down	90.2	6.3	13.4
Carry	94.3	9.1	8.3
Carry (weights)	97.2	8.7	10.3

Table 6.1: Success rate, average execution time, and average precision for all tasks. All metrics are averaged over 4096 trails per task.

objective. We consider sitting to be successful if the character hip is within 20 cm of the target location. Similarly, we declare lying down to be successful if the hip and the head of the character are both within 30 cm from a target location. The carry task is successful if the box is within 20 cm of the target location. All tasks are considered unsuccessful if their success criterion is not met within 20 seconds. We evaluate the sit and lie down tasks on 16 and 5 unseen objects respectively. To increase the variability between the objects, we randomly scale the objects in each trial with a scale factor between 0.8 and 1.2. For the carry task, we randomly scale the original box shown in the human demonstration by a scale factor between 0.5 and 1.5 in each trail. The original box is of size $50 \times 35 \times 30$ cm. The character is randomly initialized anywhere between 1 m and 10 m away from the object and with a random orientation. In addition to the success rate, we also measure the average execution time and precision for all successful trails. Execution time is the average time until the character succeeds in executing the task, according to the success definition above. Precision is the average distance between the hip, head, box and their target locations for sit, lie down, and carry respectively. All metrics are evaluated over 4096 trails per task. Similarly, we evaluate our carry policy, which is trained to carry boxes of the same size but different weights, in Table 6.1 using the same metrics. Despite the diversity of test objects and configurations, our policies succeed in executing all task with higher than 90% success rate.

We illustrate the plausibility of the full motion trajectories generated by our policy in Fig. 6.9. We initialize our policy with the first frames of the reference motion clips. We then plot the full trajectories followed by our policy alongside the reference trajectories from the reference motion clips. For the sit and lie down tasks, we plot the character trajectory. For the carry task, we plot the box trajectory. Although the reference clips do not always follow the shortest trajectory to the object, our policy often does, as can be seen in Fig. 6.9(a). Moreover, our character learns to go beyond the limited reference clips and succeeds in executing the tasks from initial configurations not shown in the reference motion as can be seen in Fig. 6.10. In the reference clips, the character starts up to three meters away from the object, nonetheless, our policy learns to execute the tasks even when initialized up to ten meters away from the object. This is due to our randomization approach in training the policy as described in Sec. 6.5.

Next we study the robustness of our policy to external perturbations. We pelt the character with 20 projectiles of weight 1.2 kg at random time steps of the episode. We found that our policy is very robust to these perturbations. The character learns to get

Task	Success Rate (%)
Sit	87.5
Lie down	82.0
Carry	89.4

Table 6.2: Success rate under physical perturbations.

Task	Success Rate (%)	Execution Time (Seconds)	Precision (cm)
Sit	88.6	5.3	6.6
Lie down	81.9	6.2	14.7
Carry	0.0	-	-

Table 6.3: Bounding box ablation. Success rate, average execution time, and average precision for all tasks.

back on track and resume the task execution upon being hit by a projectile. We also move the object during the execution of the motion (e.g. move the chair away as the character is about to sit). The policy is robust to such changes in the environment; the character quickly adapts. Our policies maintain a high success rate under the physical perturbations for all three tasks as reported in Table 6.2

Throughout our experiments, we include the bounding box of the object in our goal $\mathbf{g}_t$ as explained in Sec. 6.3. To evaluate the importance of the bounding box, we retrain our policies without this information and evaluate the policies in Table 6.3. We observe that the bounding box is vital especially for dynamic tasks like carry. Without this information, the character fails to pick up the box from the platform. In general, excluding the bounding box information decreases the success rate for all three tasks.

6.6.1 Comparisons

There are only a few previous attempts in the area of synthesizing character-scene interactions. We compare our physics-based model to NSM [188] and SAMP [74] (Chapter 5), which are both kinematic models. We also compare to Chao et al. [31], which is a hierarchical-based physical approach. All three methods are trained on the sitting task. Kinematic models (NSM and SAMP) tend to produce nonphysical behaviors, such as foot-skating/floating and object penetrations. This hinders them from generalizing to new scenarios. The work of Chao et al. [31] synthesizes physical motion, however, it often fails to sit on the target object. Most of the time, the character falls when approaching the object.

A quantitative comparison to previous methods is available in Table 6.4. A trial is considered successful, only if the character does not penetrate the object while approaching it. None of the baselines are capable of consistently completing the full carry task. NSM [188] trains a character to walk towards a box and lift it up. The character, however, needs to be manually controlled to carry the box to a destination. Our policy, on the other hand, enables the character to autonomously walk towards a box, lift

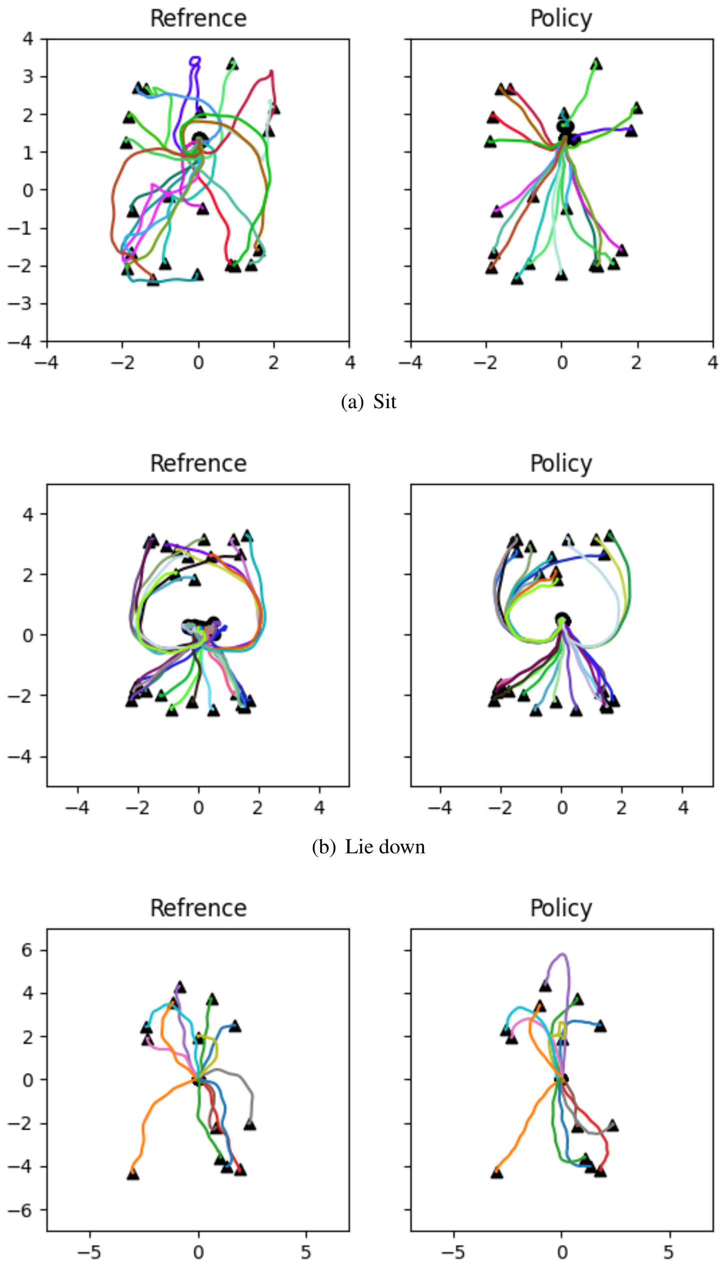
Metric	**Sit**				**Lie down**	
	NSM	SAMP	Chao et al.	Ours	SAMP	Ours
Success Rate(%)	75.0	75.0	17	**93.7**	50	**80**
Execution Time(seconds)	7.5	7.2	-	**3.7**	9.5	**6.9**
Precision (meters)	0.19	**0.06**	-	0.09	**0.05**	0.3

Table 6.4: Comparison to NSM [188], SAMP [74], Chao et al. [31]

the box, and *carry* it to the destination. We use the pre-trained open-source models of NSM [188], and SAMP [74], and evaluate them on the same test objects as our method. For Chao et al. [31], we report the numbers provided in the paper. Table 6.4 shows that our method significantly outperforms all baselines.

6.7 Conclusion

We presented a method that realistically synthesizes physical and realistic character-scene interaction. We applied our method to three challenging scene interaction tasks: sit, lie down, and carry. Our method learns when and where to transition from one behavior to another to execute the desired task. We introduced an efficient randomization approach for the training objects, their placements, sizes, and physical properties. This randomization approach allows our policies to generalize to a wide range of objects and scenarios not shown in the human demonstration. We showed that our policies are robust to different physical perturbations and sudden changes in the environment. We qualitatively and quantitatively showed that our method significantly outperforms previous work. In summary, our method is a critical step toward creating physically simulated characters that can interact realistically with their environments.

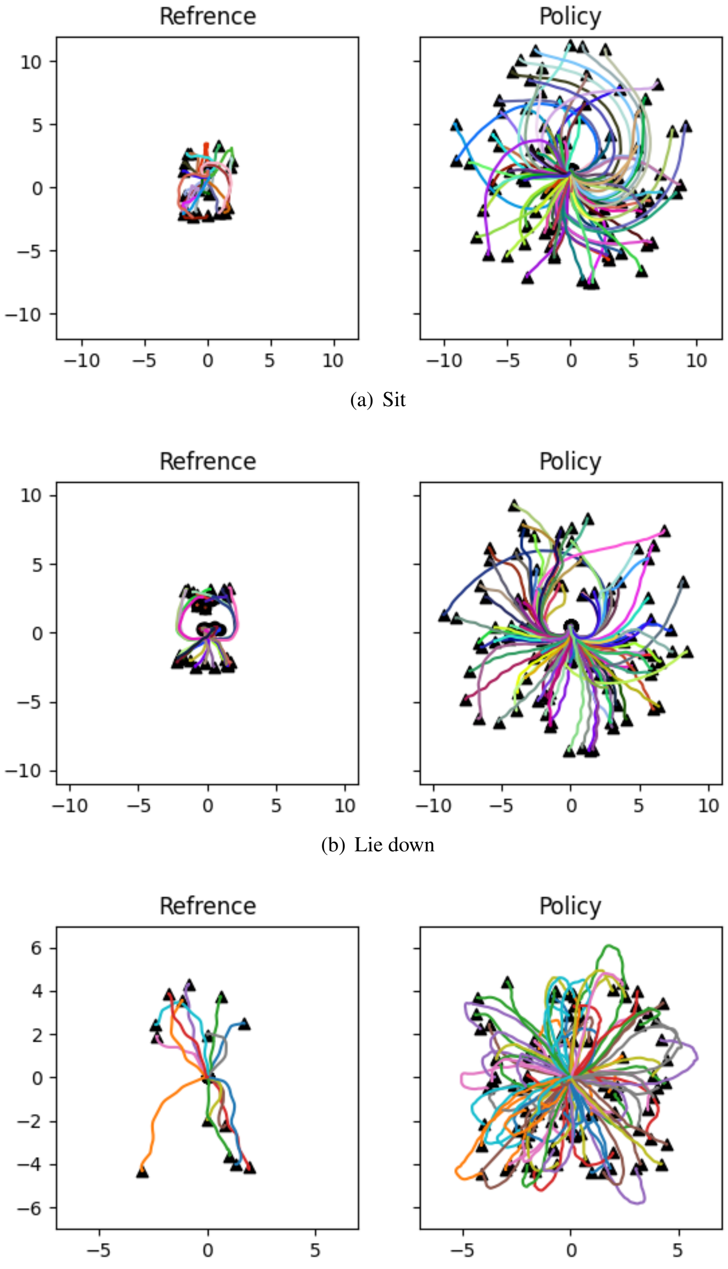


(a) Sit

(b) Lie down

(c) Carry

Figure 6.9: Reference motion trajectories and the trajectories generated by our policies when initialized with the first frame of the reference motion. Triangles indicate starting positions and the target position is indicated with a circle. Although the reference clips do not always follow the shortest trajectory to the object, our policy often does.



(a) Sit

(b) Lie down

(c) Carry

Figure 6.10: Reference motion trajectories and the trajectories generated by our policies when initialized randomly. Triangles indicate starting positions and the target position is indicated with a circle. From limited ground-truth clips covering limited environment configurations, our policy learns to successfully execute the actions in wide range of configurations.

Chapter 7

Conclusion and Future Work

In this thesis, we argue that human motion makes little sense in isolation. Humans never move in a vacuum and they are in constant interaction with the surrounding scene. Similarly, scenes are designed with human motion in mind. They are designed to offer certain affordances to humans. This thesis presents several steps toward the joint understanding of human motion and the surrounding scene. Moreover, we show that studying the human and scene jointly is necessary and useful for several applications. We focus on two interconnected problems: reconstructing and synthesizing human-scene interaction (HSI). We tackle both problems by jointly considering the human and the surrounding scene.

Our first work, PROX, focuses on reconstructing HSI. PROX shows that incorporating scene constraints in an optimization-based pose estimation system leads to significantly more realistic and accurate reconstruction. In addition, PROX created a dataset of 3D humans interacting realistically with 3D scenes.

Traditional body models, like SMPL-X, only model the human body pose and shape. We learn a new HSI model which upgrades traditional body models to explicitly model HSI. We call this model POSA and we learn it from the PROX dataset. POSA encodes contact and semantic relationships between the body and the scene. We use POSA to automate populating 3D scenes with 3D people. Given a scan of a person with a known pose, POSA allows us to search the scene for locations where the pose is likely. In addition, we show that the HSI prior of POSA improves pose estimation.

POSA enables synthesizing static HSI. We transition to address the more challenging task of synthesizing dynamic HSI. Creating virtual humans that move and act like real people, however, is challenging and requires tackling many smaller but difficult problems. Our model, SAMP, enables virtual characters to realistically navigate cluttered indoor environments. In addition, SAMP models the stochastic nature of HSI and synthesizes the same action in different styles.

Kinematic models, like SAMP, can produce high-quality motion when applied in environments similar to those shown in the dataset. However, when applied to new scenarios, kinematic models can struggle to generate realistic behaviors that respect scene constraints. We address this by InterPhys which is a method capable of synthesizing physical and life-like HSI. InterPhys leverages an adversarial discriminator to differentiate between motions in the dataset and motions generated by the RL policy. The policy is trained to satisfy a task reward while also trying to fool the discriminator by producing motions that resemble those shown in the dataset. The discriminator and the policy are both conditioned on the scene context. We introduce an efficient randomization approach for the training objects, their placements, sizes, and physical properties. This randomization approach allows our policies to generalize to a wide range of objects and scenarios not shown in the human demonstration.

Future Work

In PROX, we show how to improve pose estimation using scene information. Future work should explore how to use the human pose to improve scene reconstruction. We also assume that the scene reconstruction is given. It would be interesting to explore the joint reconstruction of humans and the surrounding scene from a single image.

Throughout this thesis, we focus on a single human interacting with a scene. Future work should explore multiple people interacting with each other and with the surrounding scene. SAMP and InterPhys are capable of synthesizing high-quality motion of a couple of skills. In the future, we hope to see methods capable of synthesizing hundreds of different skills. Moreover, SAMP and InterPhys require high-quality MoCap for training. It will be interesting if future work can relax this requirement and learn from the abundant internet videos directly.

In summary, this thesis presents four key steps toward realistic reconstruction and synthesis of the interaction between humans and the surrounding scene. We hope this thesis will spur more research in this domain.

Bibliography

[1] Kinect for xbox one. https://en.wikipedia.org/wiki/Kinect#Kinect_for_Xbox_One_(2013). 28, 33

[2] Monocle: Kinect data capture app. https://github.com/bmabey/monocle. 28

[3] Skanect: 3d scanning. https://skanect.occipital.com. 28, 33

[4] Kfir Aberman, Peizh Uo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392462. URL https://doi.org/10.1145/3386569.3392462. 22

[5] Vida Adeli, Mahsa Ehsanpour, Ian Reid, Juan Carlos Niebles, Silvio Savarese, Ehsan Adeli, and Hamid Rezatofighi. Tripod: Human trajectory and pose dynamics forecasting in the wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 23

[6] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *2009 IEEE 12th International Conference on Computer Vision*, pages 72–79, 2009. doi: 10.1109/ICCV.2009.5459148. 19

[7] Shailen Agrawal and Michiel van de Panne. Task-based locomotion. *ACM Trans. Graph.*, 35(4), 2016. 23

[8] Eren Erdal Aksoy, Alexey Abramov, Florentin Wörgötter, and Babette Dellen. Categorizing object-action relations from semantic scene graphs. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 398–405, 2010. 19

[9] Rami Ali Al-Asqhar, Taku Komura, and Myung Geol Choi. Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, page 45–53, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321327. doi: 10.1145/2485895.2485905. URL https://doi.org/10.1145/2485895.2485905. 72

[10] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 24

[11] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics interface*, volume 96, pages 222–229. Toronto, Canada, 1996. 22

[12] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of PEople. *Transactions on Graphics (TOG), Proceedings SIGGRAPH*, 24(3):408–416, 2005. 19, 29, 46

[13] Luca Ballan, Aparna Taneja, Juergen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *The European Conference on Computer Vision (ECCV)*, pages 640–653, 2012. 31

[14] Ezer Bar-Aviv and Ehud Rivlin. Functional 3D object classification using simulation of embodied agent. In *British Machine Vision Conference (BMVC)*, pages 307–316, 2006. 21

[15] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA, 2015. MIT Press. 23, 70

[16] Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. Drecon: Data-driven responsive control of physics-based characters. *ACM Trans. Graph.*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356536. URL `https://doi.org/10.1145/3355089.3356536`. 24

[17] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992. 28

[18] BMW. BMW Gesture Control — BMW Driver's Guide, 11 2021. URL `https://www.youtube.com/watch?v=Ilh9w_K7LHE&feature=youtu.be`. 14

[19] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *The European Conference on Computer Vision (ECCV)*, 2016. 20, 29, 30

[20] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5573–5582, 2017. 19

[21] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3D morphable models: Spiral convolutional networks for 3D shape representation learning and generation. In *International Conference on Computer Vision (ICCV)*, pages 7212–7221, 2019. 50

[22] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *European Conference on Computer Vision (ECCV)*, volume 12358, pages 361–378, 2020. 21

[23] Marcus A. Brubaker, David J. Fleet, and Aaron Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 87(1):140, Aug 2009. ISSN 1573-1405. doi: 10.1007/s11263-009-0274-5. URL `https://doi.org/10.1007/s11263-009-0274-5`. 19

[24] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 14, 30

[25] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision (ECCV)*, pages 387–404. Springer, 2020. 24, 65, 81

[26] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision (ECCV)*, volume 12346, pages 387–404, 2020. 21

[27] Carnegie Mellon University. CMU MoCap Dataset. URL http://mocap.cs.cmu.edu. 65

[28] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR, 30 Oct–01 Nov 2020. URL http://proceedings.mlr.press/v100/chai20a.html. 24

[29] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, pages 667–676, 2017. 14, 50

[30] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 72, 91

[31] Yu-Wei Chao, Jimei Yang, Weifeng Chen, and Jia Deng. Learning to sit: Synthesizing human-chair interactions via hierarchical control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. 13, 25, 97, 98

[32] Long Chen, Shaobo Lin, Xiankai Lu, Dongpu Cao, Hangbin Wu, Chi Guo, Chun Liu, and Fei-Yue Wang. Deep neural network based vehicle and pedestrian detection for autonomous driving: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 22 (6):3234–3246, 2021. 14

[33] Yixin Chen, Siyuan Huang, Tao Yuan, Yixin Zhu, Siyuan Qi, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3D holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *International Conference on Computer Vision (ICCV)*, pages 8647–8656, 2019. 21

[34] Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. Physics-based motion capture imitation with deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, MIG '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360159. doi: 10.1145/3274247.3274506. URL https://doi.org/10.1145/3274247.3274506. 24

[35] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J. Black. Monocular expressive body regression through body-driven attention. In *European Conference on Computer Vision (ECCV)*, pages 20–40, 2020. URL https://expose.is.tue.mpg.de. 14, 19

[36] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6992–7001, 2020. 23

[37] Enric Corona, Albert Pumarola, Guillem Alenyà, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6990–6999, 2020. 21

[38] Enric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Grégory Rogez. GanHand: Predicting human grasp affordances in multi-object scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5030–5040, 2020. 21

[39] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Generalized biped walking control. *ACM Trans. Graph.*, 29(4), jul 2010. ISSN 0730-0301. doi: 10.1145/1778765.1781156. URL https://doi.org/10.1145/1778765.1781156. 25

[40] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 14

[41] Vincent Delaitre, David F Fouhey, Ivan Laptev, Josef Sivic, Abhinav Gupta, and Alexei A Efros. Scene semantics from long-term observation of people. In *The European Conference on Computer Vision (ECCV)*, pages 284–298, 2012. 19

[42] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *Int. J. Comput. Vis.*, 61(2):185–205, 2005. doi: 10.1023/B:VISI.0000043757.18370.9c. URL https://doi.org/10.1023/B:VISI.0000043757.18370.9c. 15

[43] P Kingma Diederik and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations ICLR*, 2014. 67

[44] D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 0047-259X. doi: https://doi.org/10.1016/0047-259X(82)90077-X. URL https://www.sciencedirect.com/science/article/pii/0047259X8290077X. 79

[45] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366—-2374, 2014. 27, 45

[46] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. In *International Conference on Learning Representations ICLR*, 2014. 23

[47] Haegwang Eom, Daseong Han, Joseph S. Shin, and Junyong Noh. Model predictive control with a visuomotor system for physics-based character animation. *ACM Trans. Graph.*, 39(1), October 2019. ISSN 0730-0301. doi: 10.1145/3360905. URL https://doi.org/10.1145/3360905. 25

[48] Yao Feng, Vasileios Choutas, Timo Bolkart, Dimitrios Tzionas, and Michael Black. Collaborative regression of expressive bodies using moderation. In *International Conference on 3D Vision (3DV)*, pages 792–804, December 2021. doi: 10.1109/3DV53792.2021.00088. 14, 19

[49] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG)*, 34(6):179, 2015. 21

[50] David F Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A Efros, Ivan Laptev, and Josef Sivic. People watching: Human actions as a cue for single view geometry. *International Journal of Computer Vision (IJCV)*, 110(3):259–274, 2014. 19

[51] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 4346–4354, USA, 2015. IEEE Computer Society. ISBN 9781467383912. 23

[52] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *European conference on computer vision*, pages 368–381. Springer, 2010. 19

[53] Dariu M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding (CVIU)*, 73(1):82 – 98, 1999. 19

[54] Stuart Geman and Donald E. McClure. Statistical methods for tomographic image reconstruction. In *Proceedings of the 46th Session of the International Statistical Institute, Bulletin of the ISI*, volume 52, 1987. 30, 31

[55] James J Gibson. *The perception of the visual world*. Houghton Mifflin, 1950. 26

[56] James J Gibson. The theory of affordances. *Hilldale, USA*, 1:2, 1977. 19

[57] Michael Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, page 139–ff., New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918843. doi: 10.1145/253284.253321. URL https://doi.org/10.1145/253284.253321. 22, 84

[58] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, page 33–42, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 0897919998. doi: 10.1145/280814.280820. URL https://doi.org/10.1145/280814.280820. 22

[59] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. SpiralNet++: A fast and highly efficient mesh convolution operator. In *International Conference on Computer Vision Workshops (ICCVw)*, pages 4141–4148, 2019. 50

[60] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1536, 2011. 21, 52, 55

[61] F. Sebastin Grassia. Practical parameterization of rotations using the exponential map. *J. Graph. Tools*, 3(3):29–48, March 1998. ISSN 1086-7651. doi: 10.1080/10867651.1998.10487493. URL https://doi.org/10.1080/10867651.1998.10487493. 88

[62] Renshu Gu, Gaoang Wang, and Jenq-Neng Hwang. Efficient multi-person hierarchical 3d pose estimation for autonomous driving. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 163–168. IEEE, 2019. 14

[63] Abhinav Gupta, Trista Chen, Francine Chen, Don Kimber, and Larry S Davis. Context and observation driven latent variable model for human pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 8, 2008. doi: 10.1109/CVPR.2008.4587511. 20

[64] Abhinav Gupta, Aniruddha Kembhavi, and Larry S Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(10):1775–1789, 2009. 19

[65] Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. From 3d scene geometry to human workspace. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1961–1968, 2011. 19, 21

[66] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2255–2264, 2018. 24

[67] I. Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and T. Komura. A recurrent variational autoencoder for human motion synthesis. In *BMVC*, 2017. 23

[68] Christian Häne, Christopher Zach, Andrea Cohen, and Marc Pollefeys. Dense semantic 3d reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 39 (9):1730–1743, 2016. 14

[69] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107, 1968. doi: 10.1109/TSSC.1968.300136. 70

[70] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392480. URL https://doi.org/10.1145/3386569.3392480. 22

[71] Nils Hasler, Bodo Rosenhahn, Thorsten Thormahlen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–231, June 2009. doi: 10.1109/CVPR.2009.5206859. 20

[72] Nils Hasler, Carsten Stoll, Martin Sunkel, Bodo Rosenhahn, and Hans-Peter Seidel. A statistical model of human pose and body shape. *Computer Graphics Forum*, 28(2):337–346, 2009. 29

[73] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *International Conference on Computer Vision*, pages 2282–2292, October 2019. URL https://prox.is.tue.mpg.de. 12, 16, 19, 20, 47, 49, 55, 59, 61, 63

[74] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proceedings of the International Conference on Computer Vision 2021*, October 2021. 13, 17, 23, 24, 84, 90, 97, 98

[75] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 16, 19, 21

[76] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 65

[77] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. In *preprint*, 2022. 25

[78] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 20, 21

[79] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 14

[80] Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017. 84

[81] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. MoGlow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Trans. Graph.*, 39(6), November 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417836. URL https://doi.org/10.1145/3414685.3417836. 24

[82] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017. 50

[83] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf. 25

[84] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4):1–11, 2016. 23

[85] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Trans. Graph.*, 36(4), July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073663. URL https://doi.org/10.1145/3072959.3073663. 23, 65, 77, 84

[86] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392440. URL https://doi.org/10.1145/3386569.3392440. 22

[87] Shiyu Huang, Wenze Chen, Longfei Zhang, Ziyang Li, Fengming Zhu, Deheng Ye, Ting Chen, and Jun Zhu. Tikick: Towards playing multi-agent football full games from single-agent demonstrations. *arXiv preprint arXiv:2110.04507*, 2021. 25

[88] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 34–50, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46466-4. 14, 15

[89] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2014. 19

[90] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79. 23

[91] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. 2015. 32

[92] Yun Jiang, Hema Koppula, and Ashutosh Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2993–3000, 2013. 21

[93] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986. 23

[94] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8320–8329, 2018. 14, 19, 46

[95] Shanon X. Ju, Michael J. Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, FG '96, page 38, USA, 1996. IEEE Computer Society. ISBN 0818677139. 15

[96] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 19

[97] Changgu Kang and Sung-Hee Lee. Environment-adaptive contact poses for virtual characters. *Computer Graphics Forum (CGF)*, 33(7):1–10, 2014. 21

[98] Mubbasir Kapadia, Xu Xianghao, Maurizio Nitti, Marcelo Kallmann, Stelian Coros, Robert W. Sumner, and Markus Gross. Precision: Precomputing environment semantics for contact-rich character animation. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, page 29–37, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340434. doi: 10.1145/2856400.2856404. URL https://doi.org/10.1145/2856400.2856404. 23

[99] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. *arXiv preprint arXiv:2010.11531*, 2020. 22

[100] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics (TOG)*, 33 (4):120, 2014. 20, 52, 55

[101] Y. Kim, H. Park, S. Bang, and S. Lee. Retargeting human-object interaction to virtual avatars. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2405–2412, 2016. doi: 10.1109/TVCG.2016.2593780. 22

[102] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 51, 73

[103] Hedvig Kjellström, Danica Kragić, and Michael J Black. Tracking people interacting with objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 747–754, 2010. 19

[104] Hedvig Kjellström, Javier Romero, and Danica Kragić. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding (CVIU)*, 115(1):81–90, 2011. 52

[105] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4501–4510, 2019. 51

[106] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 19

[107] Arthur D Kuo. A simple model of bipedal walking predicts the preferred speed–step length relationship. *Journal of biomechanical engineering*, 123(3):264–269, 2001. 20

[108] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3D scene tracking: The single actor hypothesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9–16, 2013. 20

[109] Nikolaos Kyriazis and Antonis Argyros. Scalable 3D tracking of multiple interacting objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3430–3437, 2014. 20, 26

[110] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 39–48, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311539. URL https://doi.org/10.1145/311535.311539. 22, 84

[111] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.*, 21 (3):491–500, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566607. URL `https://doi.org/10.1145/566654.566607`. 22, 23

[112] Kang Hoon Lee, Myung Geol Choi, and Jehee Lee. Motion patches: Building blocks for virtual environments annotated with motion data. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, page 898–906, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933646. doi: 10.1145/1179352.1141972. URL `https://doi.org/10.1145/1179352.1141972`. 22, 23

[113] Kurt Leimer, Andreas Winkler, Stefan Ohrhallinger, and Przemyslaw Musialski. Pose to seat: Automated design of body-supporting surfaces. *Computer Aided Geometric Design (CAGD)*, 79:101855, 2020. 21

[114] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*, 2020. 24

[115] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Learn to dance with AIST++: Music conditioned 3D dance generation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 24

[116] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3D indoor environments. In *Computer Vision and Pattern Recognition (CVPR)*, pages 12368–12376, 2019. 21

[117] Zongmian Li, Jiri Sedlar, Justin Carpentier, Ivan Laptev, Nicolas Mansard, and Josef Sivic. Estimating 3d motion and forces of person-object interactions from monocular video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 20

[118] Juncong Lin, Takeo Igarashi, Jun Mitani, Minghong Liao, and Ying He. A sketching interface for sitting pose design in the virtual environment. *Transactions on Visualization and Computer Graphics (TVCG)*, 18(11):1979–1991, 2012. 20

[119] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392422. URL `https://doi.org/10.1145/3386569.3392422`. 23, 24, 65, 68, 70, 77

[120] Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Trans. Graph.*, 37(4), jul 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201315. URL `https://doi.org/10.1145/3197517.3201315`. 25

[121] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, SM Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, et al. From motor control to team play in simulated humanoid football. *arXiv preprint arXiv:2105.12196*, 2021. 25

[122] Zhenbao Liu, Caili Xie, Shuhui Bu, Xiao Wang, Junwei Han, Hongwei Lin, and Hao Zhang. Indirect shape analysis for 3D shape retrieval. *Computers & Graphics (CG)*, 46: 110–116, 2015. 21

[123] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *Transactions on Graphics (TOG), Proceedings SIGGRAPH Asia*, 34(6):248:1–248:16, 2015. 19, 29, 46

[124] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision (ICCV)*, pages 5441–5450, 2019. 12, 19, 28, 33, 34, 35

[125] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5442–5451, October 2019. 65, 70

[126] O. Makansi, E. Ilg, Ö. Çiçek, and T. Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL http://lmb.informatik.uni-freiburg.de/Publications/2019/MICB19. 24

[127] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 91

[128] Priyanka Mandikal, Navaneet KL, and R Venkatesh Babu. 3d-psrnet: Part segmented 3d point cloud reconstruction from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 14

[129] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 65

[130] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2891–2900, 2017. 23

[131] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44:1–44:14, July 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073596. URL http://doi.acm.org/10.1145/3072959.3073596. 14, 19

[132] Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Catch & carry: Reusable neural controllers for vision-guided whole-body tasks. *ACM Trans. Graph.*, 39(4), July 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392474. URL https://doi.org/10.1145/3386569.3392474. 25

[133] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3481–3490, Stockholmsmässan, Stockholm

Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/mescheder18a.html. 88

[134] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 14

[135] Thomas B. Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding (CVIU)*, 81(3):231–268, 2001. 14, 19

[136] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2):90–126, 2006. 19

[137] Aron Monszpart, Paul Guerrero, Duygu Ceylan, Ersin Yumer, and Niloy J Mitra. imapper: interaction-guided scene mapping from monocular videos. *ACM Transactions on Graphics (TOG)*, 38(4):92, 2019. 21

[138] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.*, 31(4), jul 2012. ISSN 0730-0301. doi: 10.1145/2185520.2185539. URL https://doi.org/10.1145/2185520.2185539. 25

[139] Lucas Mourot, Ludovic Hoyet, François Le Clerc, François Schnitzler, and Pierre Hellier. A survey on deep learning for skeleton-based human animation. In *Computer Graphics Forum*. Wiley Online Library, 2021. 25

[140] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *International Conference on Computer Vision (ICCV)*, pages 2901–2910, 2019. 21

[141] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database HDM05. Technical Report CG-2007-2, Universität Bonn, June 2007. 65

[142] Muzammal Naseer, Salman Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. *IEEE Access*, 7:1859–1887, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2886133. 27, 45

[143] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 14

[144] Jorge Nocedal and Stephen J Wright. *Nonlinear Equations*. Springer, 2006. 32

[145] Occipital. Structure sensor: 3d scanning, augmented reality and more. https://structure.io/structure-sensor. 28, 33

[146] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 2088–2095, 2011. 20, 26

[147] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, September 2018. 19

[148] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: Sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, volume 12351, pages 598–613, 2020. 19

[149] Sang Il Park, Hyun Joon Shin, and Sung Yong Shin. On-line locomotion generation based on motion blending. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, page 105–111, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135734. doi: 10.1145/545261.545279. URL https://doi.org/10.1145/545261.545279. 22

[150] Priyanka Patel, Chun-Hao Paul Huang, Joachim Tesch, David Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 47, 55, 57, 59, 62

[151] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 14

[152] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 19

[153] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019. 15, 16, 19, 21, 26, 27, 29, 30, 31, 32, 34, 43, 45, 46, 48, 55, 63, 70

[154] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 14

[155] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4), jul 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201311. URL https://doi.org/10.1145/3197517.3201311. 24, 84, 91

[156] Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. *MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies*. Curran Associates Inc., Red Hook, NY, USA, 2019. 25

[157] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459670. URL https://doi.org/10.1145/3450626.3459670. 25, 84, 86, 88, 92

[158] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2883–2896, Dec 2018. 20, 26

[159] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2847–2854, 2012. 19

[160] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Björn Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937. IEEE, June 2016. doi: 10.1109/CVPR.2016.533. 14, 15

[161] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding (CVIU)*, 108(1-2):4–18, 2007. 19

[162] Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *International Conference on Computer Vision (ICCV)*, pages 4331–4340, 2019. 21, 59

[163] Marc H. Raibert and Jessica K. Hodgins. Animation of dynamic legged locomotion. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '91, page 349–358, New York, NY, USA, 1991. Association for Computing Machinery. ISBN 0897914368. doi: 10.1145/122718.122755. URL `https://doi.org/10.1145/122718.122755`. 24

[164] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, volume 11207, pages 725–741, 2018. 50

[165] Kathleen M. Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, Scott Fleming, Tina Brill, David Hoeferlin, and Dennis Burnsides. Civilian American and European Surface Anthropometry Resource (CAESAR) final report. Technical Report AFRL-HE-WP-TR-2002-0169, US Air Force Research Laboratory, 2002. 29

[166] Grégory Rogez, James S. Supančič III, and Deva Ramanan. Understanding everyday hands in action from rgb-d images. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 3889–3897, 2015. 20, 26

[167] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *Transactions on Graphics (TOG), Proceedings SIGGRAPH Asia*, 36(6):245:1–245:17, 2017. 21

[168] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017. 29

[169] Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, September 1998. ISSN 0272-1716. doi: 10.1109/38.708559. URL `https://doi.org/10.1109/38.708559`. 22

[170] Bodo Rosenhahn, Christian Schmaltz, Thomas Brox, Joachim Weickert, Daniel Cremers, and Hans-Peter Seidel. Markerless motion capture of man-machine interaction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587520. 19

[171] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 24

[172] Martin Rünz and Lourdes Agapito. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4471–4478, 2017. 45

[173] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 24

[174] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, page 514–521, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 9781450378239. doi: 10.1145/1186562.1015754. URL https://doi.org/10.1145/1186562.1015754. 23

[175] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding (CVIU)*, 152:1–20, 2016. ISSN 1077-3142. 19

[176] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. Scenegrok: Inferring action maps in 3d environments. *ACM Transactions on graphics (TOG)*, 33(6):212, 2014. 21

[177] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics (TOG)*, 35(4):139, 2016. 8, 21, 32, 33, 35, 38, 42

[178] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347. 92

[179] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne P. Tchapmi, Micael E. Tchapmi, Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. iGibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, page accepted. IEEE, 2021. 65

[180] Hubert P. H. Shum, Taku Komura, Masashi Shiraishi, and Shuntaro Yamazaki. Interaction patches for multi-character animation. *ACM Trans. Graph.*, 27(5), December 2008. ISSN 0730-0301. doi: 10.1145/1409060.1409067. URL https://doi.org/10.1145/1409060.1409067. 23

[181] H. Sidenbladh, M. J. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *European Conf. on Computer Vision (ECCV)*, volume 1, pages 784–800, 2002. 24

[182] L. Sigal, A. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(4):4–27, March 2010. doi: 10.1007/s11263-009-0273-6. 65

[183] Leonid Sigal, Alexandru Balan, and Michael J Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision (IJCV)*, 87(1-2):4–27, 2010. 19

[184] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 30

[185] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015. 67

[186] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *Computer Vision and Pattern Recognition (CVPR)*, pages 190–198, 2017. 21

[187] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *The European Conference on Computer Vision (ECCV)*, pages 294–310, 2016. 20

[188] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356505. URL `https://doi.org/10.1145/3355089.3356505`. 10, 13, 23, 68, 72, 74, 76, 77, 78, 79, 84, 97, 98

[189] Jan Stenum, Kendra M. Cherry-Allen, Connor O. Pyles, Rachel D. Reetzke, Michael F. Vignos, and Ryan T. Roemmich. Applications of pose estimation in human health and performance across the lifespan. *Sensors*, 21(21), 2021. ISSN 1424-8220. doi: 10.3390/s21217315. URL `https://www.mdpi.com/1424-8220/21/21/7315`. 15

[190] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 57

[191] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, volume 12349, pages 581–600, 2020. 21

[192] Graham W. Taylor and Geoffrey E. Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1025–1032, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553505. URL https://doi.org/10.1145/1553374.1553505. 23

[193] Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Transactions on Graphics (TOG)*, 36(6):244:1–244:12, November 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130853. URL http://doi.acm.org/10.1145/3130800.3130853. 45

[194] Matthias Teschner, Stefan Kimmerle, Bruno Heidelberger, Gabriel Zachmann, Laks Raghupathi, Arnulph Fuhrmann, Marie-Paule Cani, François Faure, Nadia Magnenat-Thalmann, Wolfgang Strasser, and Pascal Volino. Collision detection for deformable objects. In *Eurographics*, pages 119–139, 2004. 31

[195] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014. doi: 10.1109/CVPR.2014.214. 15

[196] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3D human pose estimation fusing video and inertial sensors. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 14.1–14.13, September 2017. ISBN 1-901725-60-X. doi: 10.5244/C.31.14. URL https://dx.doi.org/10.5244/C.31.14. 65

[197] Aggeliki Tsoli and Antonis A. Argyros. Joint 3d tracking of a deformable object in interaction with a hand. In *The European Conference on Computer Vision (ECCV)*, 2018. 20, 26

[198] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 118(2):172–193, 2016. 20, 26, 31

[199] Munetoshi Unuma, Ken Anjyo, and Ryozo Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 91–96, New York, NY, USA, 1995. Association for Computing Machinery. ISBN 0897917014. doi: 10.1145/218380.218419. URL https://doi.org/10.1145/218380.218419. 22

[200] H. van Welbergen, B.J.H. van Basten, A. Egges, Z.M. Ruttkay, and M.H. Overmars. Real time animation of virtual humans: A trade-off between naturalness and control. *Computer graphics forum*, 29(8):2530–2554, December 2010. ISSN 0167-7055. doi: 10.1111/j.1467-8659.2010.01822.x. 22

[201] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *international conference on machine learning*, pages 3560–3569. PMLR, 2017. 23

[202] Marek Vondrak, Leonid Sigal, and Odest Chadwicke Jenkins. Dynamical simulation priors for human motion tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):52–65, Jan 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.61. 20

[203] He Wang, Sören Pirk, Ersin Yumer, Vladimir Kim, Ozan Sener, Srinath Sridhar, and Leonidas Guibas. Learning a generative model for multi-step human-object interactions from videos. *Computer Graphics Forum (CGF)*, 38(2):367–378, 2019. 21

[204] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Optimizing walking controllers. *ACM Trans. Graph.*, 28(5):1–8, dec 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618514. URL https://doi.org/10.1145/1618452.1618514. 24

[205] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3D human motion and interaction in 3D scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. 23

[206] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 14

[207] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion, 2020. 24

[208] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3366–3375, 2017. 21

[209] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021. 14

[210] Z. Wang, J. Chai, and S. Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):14–28, 2021. doi: 10.1109/TVCG.2019.2938520. 24

[211] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 30

[212] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39(4), jul 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392381. URL https://doi.org/10.1145/3386569.3392381. 24

[213] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Trans. Graph.*, 40(4), jul 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459761. URL https://doi.org/10.1145/3450626.3459761. 25

[214] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9079, 2018. 65

[215] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitio)*, June 2020. 65

[216] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6183–6192, 2020. 19

[217] Jingwei Xu, Huazhe Xu, Bingbing Ni, Xiaokang Yang, Xiaolong Wang, and Trevor Darrell. Hierarchical style-based networks for motion synthesis. In *European Conference on Computer Vision (ECCV)*, pages 178–194, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58621-8. 24

[218] Pei Xu and Ioannis Karamouzas. A gan-like approach for physics-based imitation learning and interactive character control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4(3):1–22, 2021. 25

[219] Masanobu Yamamoto and Katsutoshi Yagishita. Scene constraints-aided tracking of human body. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 151–156 vol.1, June 2000. doi: 10.1109/CVPR.2000.855813. 19

[220] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17–24, 2010. 19

[221] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision (ECCV)*, 2020. 24, 78

[222] Ye Yuan and Kris M. Kitani. Diverse trajectory forecasting with determinantal point processes. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=ryxnY3NYPS`. 24

[223] S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1177–1193, 2012. doi: 10.1109/TNNLS.2012.2200299. 23

[224] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2148–2157, 2018. 20

[225] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201366. URL `https://doi.org/10.1145/3197517.3201366`. 23, 79, 84

[226] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *European Conference on Computer Vision (ECCV)*, volume 12357, pages 34–51, 2020. 21

[227] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *International Conference on 3D Vision (3DV)*, pages 642–651, 2020. 9, 21, 47, 57, 59, 61, 62

[228] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6193–6203, 2020. 21, 22, 47, 49, 50, 55, 62

[229] Yan Zhang, Michael J. Black, and Siyu Tang. We are more than our joints: Predicting how 3D bodies move. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, page 3372–3382, June 2021. 78

[230] Tao Zhao and Ram Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(9):1208–1221, Sep. 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.73. 20

[231] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey, 2020. 19

[232] Youyi Zheng, Han Liu, Julie Dorsey, and Niloy J Mitra. Ergonomics-inspired reshaping and exploration of collections of models. *Transactions on Visualization and Computer Graphics (TVCG)*, 22(6):1732–1744, 2016. 21

[233] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 28

[234] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representations ICLR*, 2018. 23

[235] Yi Zhou, Jingwan Lu, Connelly Barnes, Jimei Yang, Sitao Xiang, et al. Generative tweening: Long-term inbetweening of 3D human motions. *arXiv preprint arXiv:2005.08891*, 2020. 22

[236] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. Inferring forces and learning human utilities from videos. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3823–3833, 2016. 21