Estimating 3D Motion and Forces from Monocular Videos

Zongmian Li

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Estimating 3D Motion and Forces from Monocular Videos

Soutenue par

Zongmian Li

Le 17 février 2023

École doctorale n°386

Sciences Mathématiques de Paris Centre

Spécialité

Informatique

Composition du jury :

Andrea Cherubini	LIRMM, Université de Montpellier	*Président du jury*
Vincent Lepetit	École des Ponts ParisTech	*Rapporteur*
Patrick Pérez	Valeo.ai	*Rapporteur*
Justin Carpentier	École Normale Supérieure, Inria	*Examinateur*
Gül Varol	École des Ponts ParisTech	*Examinateur*
Nicolas Mansard	LAAS-CNRS	*Examinateur*
Josef Sivic	École Normale Supérieure, Inria	*Directeur de thèse*
Ivan Laptev	École Normale Supérieure, Inria	*Codirecteur de thèse*

ENS | PSL

Résumé

Dans cette thèse, nous étudions le problème de la reconstruction automatique en 3D des mouvements d'une personne agissant dans une scène complexe avec un objet, à partir d'une seule vidéo RVB. Nous développons une méthode complète pour établir une correspondance entre les images vidéo 2D et une interprétation 3D de la scène, qui est représentée par les poses 3D de la personne et de l'objet manipulé, les positions des contacts avec l'objet et avec l'environnement, et les forces de contact exercées à ces interfaces. Ce problème est difficile pour de multiples raisons, en particulier, des occlusions, des ambiguïtés de profondeur et des propriétés d'apparence des objets longiligne sans texture tels que la bêche ou le marteau. Les principales contributions de cette thèse sont les suivantes. Dans un premier temps, nous introduisons une approche pour estimer conjointement le mouvement et les forces impliqués dans la vidéo en formulant un problème d'optimisation avec contrainte de trajectoire minimisant une fonction de perte, composite, intégrée dans le temps. Les variables de décision de ce problème sont les trajectoires de la personne et de l'outil qu'il manipule, ainsi que les forces d'interaction entre la personne, l'outil et l'environnement. Les variables permettent une modélisation physique de la scène, du mouvement des corps sans l'action des faces aux points de contact. Les fonctions de perte portent sur les articulations de la personne et les points clé de l'objet en cherchant à minimiser la vraisemblance des observations dans l'image. Le problème est soumis à plusieurs contraintes exprimant les lois de

la mécanique, qui incluent les modèles de contact et de frottement et l'équation dynamique lagrangienne. Deuxièmement, nous développons une méthode pour reconnaître automatiquement à partir de la vidéo d'entrée la position 2D et les instants de contact entre la personne et l'objet ou le sol. Pour ce faire, nous proposons de reconnaître automatiquement les contacts dans la vidéo d'entrée à l'aide d'un réseau neuronal convolutif (en anglais CNN) entraîné à partir de données de contact annotées manuellement qui combinent à la fois des images fixes et des vidéos récoltées sur Internet. Ainsi, au lieu de modéliser les états de contact en tant que variables binaires lors de l'optimisation, nous conservons un problème d'optimisation de trajectoire sans variable mixte binaire, d'une complexité algorithmique acceptable, tant en permettant à la reconstruction de s'adapter à des changements de contact complexes sans connaissance préalable. Troisièmement, nous validons expérimentalement notre approche sur un jeu de données vidéo-MoCap récent capturant des actions typiques de parkour et équipé de forces et de trajectoires de vérité au sol. Nous démontrons également les avantages de notre approche sur un nouvel ensemble de données de vidéos Internet montrant des personnes manipulant une variété d'outils dans des environnements sans contraintes. Les expériences montrent que notre méthode améliore les résultats à la fois sur l'estimation de la pose humaine 3D et la localisation de l'objet 2D, et réalise des estimations de force raisonnables sur ces données.

Abstract

In this thesis, we investigate the problem of automatically reconstructing the 3D dynamic scene depicting a person interacting with a tool in a single RGB video. The objective is to obtain a 3D interpretation of the scene represented by the 3D poses of the person and the manipulated object over time, the contact positions and the contact forces exerted on the human body. This problem is challenging because of occlusions, depth ambiguities and the thin, texture-less nature of the manipulated tools such as the spade or the hammer. The main contributions of this thesis are as follows. First, we introduce an approach to jointly estimate the motion and the actuation forces of the person on the manipulated object by modeling the contacts and the dynamics of the interactions. This is cast as a large-scale trajectory optimization problem by minimizing a set of loss functions integrated over time and summed over person joints and object keypoints. The problem is subject to several constraints based on the laws of physics, which include contact and friction models and the Lagrangian dynamics equation. Second, we develop a method to automatically recognize from the input video the 2D position and timing of contacts between the person and the object or the ground. Instead of modeling contact states as binary variables during optimization, we automatically recognize contacts in the input video using a convolutional neural network (CNN) trained from manually annotated contact data that combine both still images and videos harvested from the Internet, thereby significantly reducing the complexity

of the optimization. Third, we validate our approach on a recent video-MoCap dataset capturing typical parkour actions and equipped with ground truth forces and trajectories. We also demonstrate the benefits of our approach on a new dataset of Internet videos showing people manipulating a variety of tools in unconstrained environments. The experiments show that our method improves results on both 3D human pose estimation and 2D object localization, and achieves reasonable force estimates on this data.

Acknowledgements

I would first like to express my gratitude to my research supervisors, Josef Sivic, Ivan Laptev and Nicolas Mansard. Thank you for your assistance and guidance throughout my PhD journey. Thank you for always understanding me, believing in me and helping me get through hard times. This thesis would have never been accomplished without your support and encouragement.

I would also like to thank all the members of the Willow team. Special thanks to the researchers who I have directly or indirectly collaborated with: Justin Carpentier, Jiri Sedlar, Galo Maldonado, Jean-Baptiste Alayrac, Yann Labbé, Gül Varol, Antoine Miech, Ignacio Rocco and Dimitri Zhukov.

I would also like to acknowledge all the jury members for reading and examining the thesis.

Finally, I would like to thank my wife, Yawen, for her patience, encouragement and unconditional support during all these years with me in France. I am forever grateful.

Contents

Chapter 1

Introduction

People can easily learn how to complete a manipulation task by observing other people performing the task or by watching an instructional video on the Internet. For example, for renovating an overgrown yard, one can find videos online with demonstrations of how to cut grass with a scythe, how to dig in hard soil with a shovel or spade, or how to break out old concrete paths with a sledgehammer in an efficient way. By watching such videos, people can imitate and perform the tool manipulation techniques in a different context, i.e. in a different environment using tools of different sizes and models. This process of watching and imitating involves highly advanced visual intelligence capabilities. These include the recognition and the interpretation of the 3D motion and dynamics of human demonstrators from instructional videos and the manipulated objects (scythe, shovel, spade and sledgehammer in these examples), as well as the interactions that are required to achieve these tasks.

An interesting question naturally arises: Is it possible to design a computer algorithm that can automate such visual understanding capabilities? In this thesis, we attempt to seek an answer to this question.

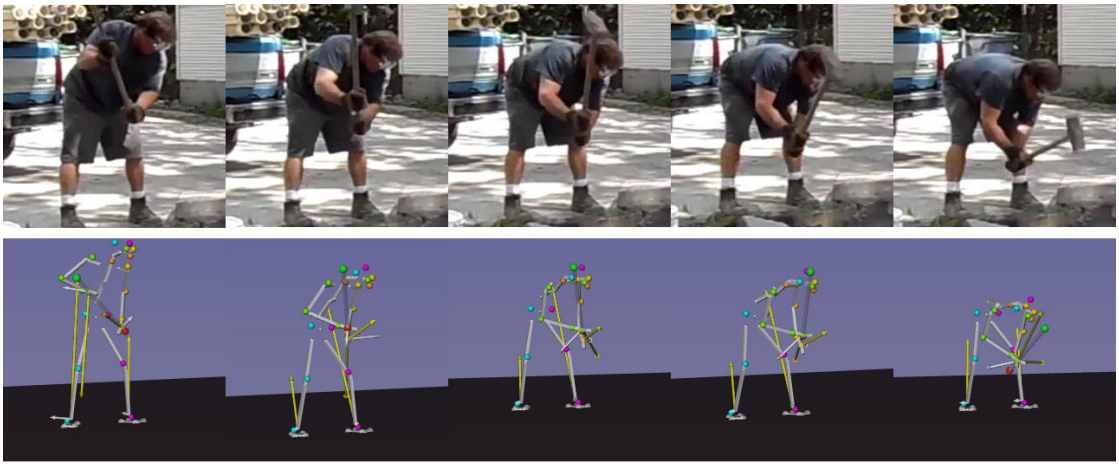


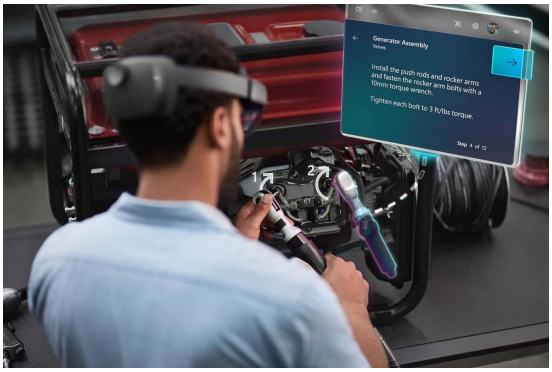
Figure 1-1: **Goal of this thesis.** We seek to develop a framework for automatic estimation of person and object 3D motion and forces, together with their interactions from a single RGB video. **Top**: An input sequence of video frames capturing the task of breaking old concrete path with a sledgehammer. **Bottom**: Output 3D motion of the person and the hammer (represented by joints and links) and the recovered contact forces (shown as yellow arrows).

1.1 Goal

The scientific goal of this thesis is to develop a framework for understanding instructional videos which involve tool manipulation. Given a single unconstrained video as input, we would like to extract information about the person-object dynamics and their interaction that are sufficient to describe the tool manipulation skill demonstrated in the video.

Modelling and understanding person-object interactions has been an active research topic in computer vision and robotics. But existing work cannot be directly applied to achieve our goal. For example, in computer vision community, most action recognition approaches usually model person-object interactions and contacts in the 2D image space but not the 3D real world space. The work from robotics and computer animation community usually considers person-object interaction in a simulated 3D space, and model the problem as an optimal control problem, which

(a) HoloLens 2, a pair of mixed reality smart glasses developed by Microsoft.



(b) Step-by-step holographic instructions for generator assembly with Hololens 2[1].

Figure 1-2: **Motivating application I: Automatic guidance via smart glasses.** The guiding instructions for manipulation actions could be learnt from instructional video data on the Internet.

is usually transcribed into a high-dimensional numerical optimization problem, seeking to minimize an objective function under contact and feasibility constraints. The contact information in such formulations is either assumed to be constant or considered as an optimization variable, which is often discrete and known to be extremely hard to optimize.

To overcome these limitations, we propose in this thesis to put the scientific tools in computer vision and robotics into a single, unified framework. More precisely, given an input RGB video, we would like to estimate the person and object 3D motion and the actuation forces directly from RGB without manually annotating the contact phases. An illustration of this goal is shown in Figure 1-1.

1.2 Motivation

Understanding complex person-object interactions from video is a key step toward the goal of building autonomous machines which can learn how to interact with

[1]Put mixed reality to work with Dynamics 365(https://dynamics.microsoft.com/en-us/mixed-reality/guides/)

Figure 1-3: **Motivating application II: Smart assistive robots.** Imagine a smart robot that is able to search for instructional video data online and to learn by itself how to accomplish a variety of object manipulation tasks. The humanoid robot *Talos*, shown in the image on the left, is 1.75 meters tall, weighs 95 kg and is able to stand on its own two legs. The image on the right shows Talos holding tools in its hands. An example of initial work in this direction demonstrating how to transfer tool manipulation skills from an instructional video to a robot is shown in Figure 1-4.

the physical world by simply observing people. These types of machines would make great changes in both industrial production and everyday life.

One application with a wide potential impact is automatic guidance. Imagine a virtual assistant in the form of smart glasses, for example, as the one shown in Figure 1-2. Such smart glasses have a wide range of applications such as (i) guiding children through simple games to improve their manipulation and language skills, (ii) helping elderly people to achieve everyday tasks, or (iii) facilitating the training of new workers in industry for highly-specialized machinery maintenance, as illustrated in Figure 1-2b.

In addition, such visual intelligence capabilities will be essential for constructing smart assistant robots that automatically learn new skills by just observing people. The concept is illustrated by two examples in Figure 1-3 and Figure 1-4. One

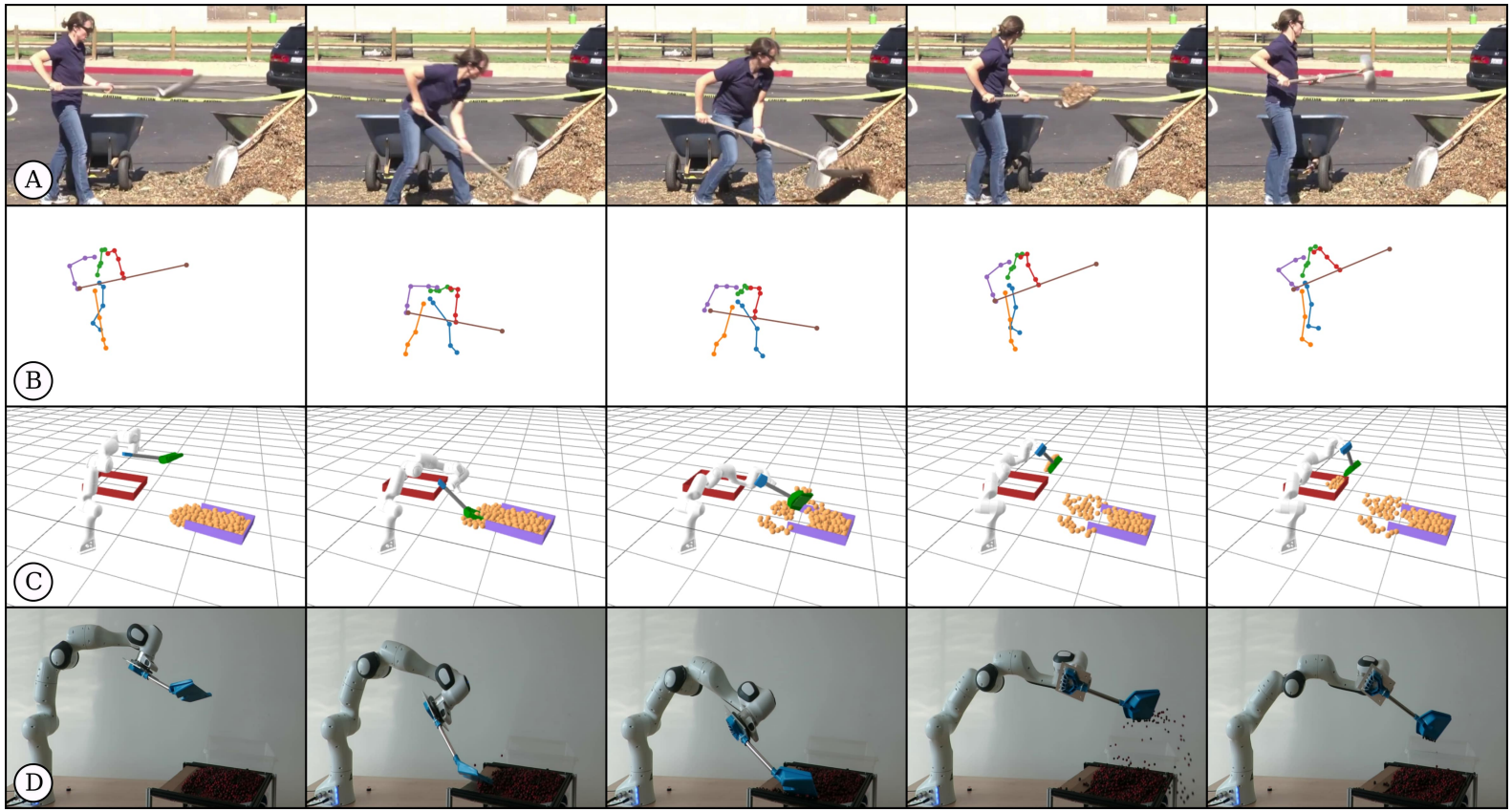


Figure 1-4: An example showing how to learn the policy of moving sand with a spade from an Internet instructional video, and then transfer the skill to the Panda robot. The rows from top to bottom shows an input video (row A), the 3D motion information extracted from the video (row B), a robot policy learnt from the 3D motion (row C) and the policy applied to the robot (row D). Figure adapted from Zorina et al. [2021].

possible solution to achieve this goal is to extract 3D information from Internet instructional videos, from which one can learn a policy which achieves a specific task and can be transferred to a robot.

In this thesis, we make a step towards these exciting applications. Our goal is to develop new models and algorithms that capture dynamic person-object interactions from instructional videos on the Internet.

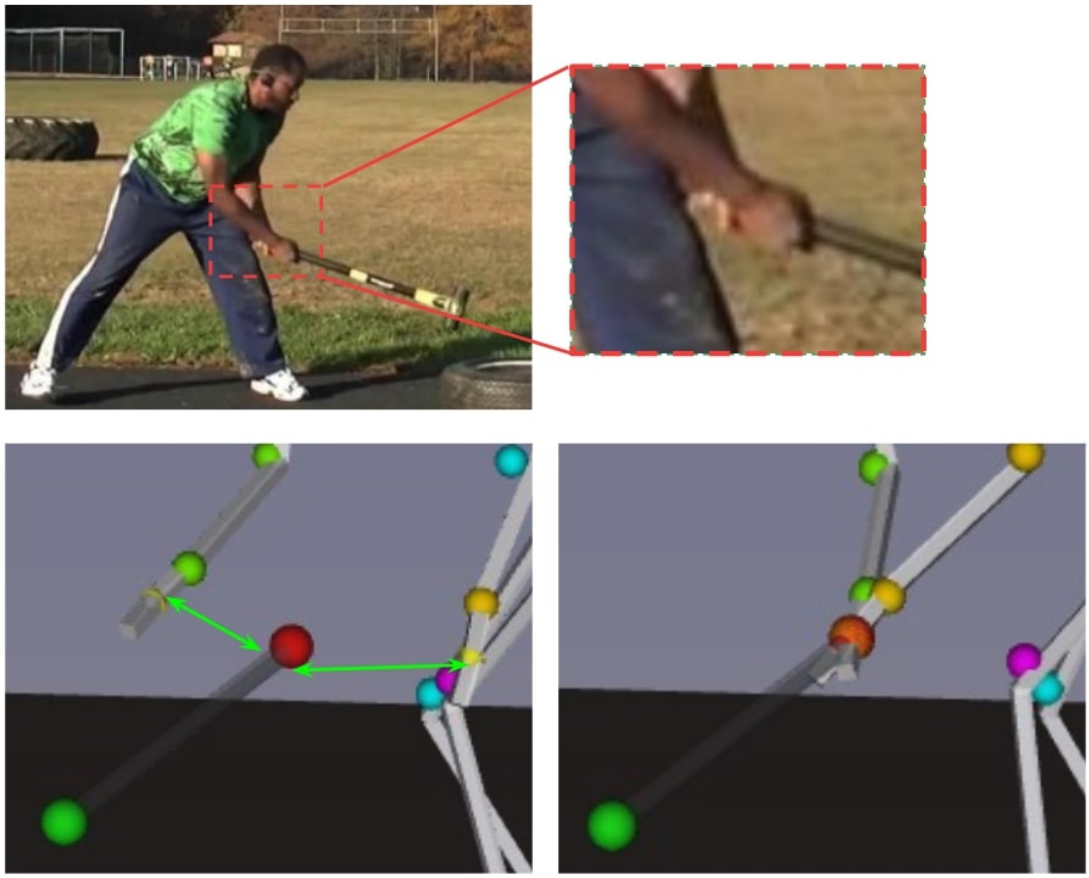


Figure 1-5: **Challenge I: Depth ambiguities.** The image on top shows an input image with close-up at the person's hands. The hands could be open or closed as shown on the bottom row. Modelling contacts with the tool can help disambiguate the two solutions.

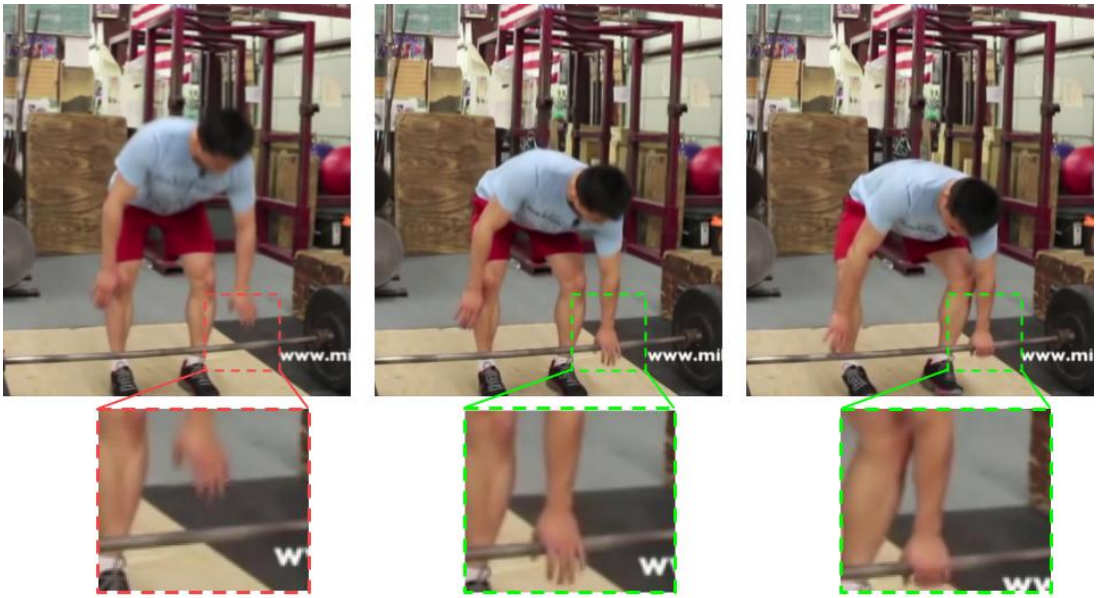


Figure 1-6: **Challenge II: Modelling and recognizing contacts.** Top row shows a sequence of frames capturing the process of a person grasping a barbell bar. Bottom row shows close-up at the person's left hand, with different color indicating different contact states (green: in contact, red: not in contact).

1.3 Challenges

The potential applications are exciting. However, there is also a range of difficult challenges and scientific problems that need to be addressed:

1. **Dealing with depth ambiguities.** As shown in Figure 1-5, there are inherent ambiguities in the 2D-to-3D mapping from a single view: multiple 3D human poses correspond to the same 2D input.

2. **Recognizing and modelling contacts.** As shown in Figure 1-6, human-object interactions often involve contacts, resulting in discontinuities in the motion of the object and the human body part in contact. For example, one must place a hand on the hammer handle before picking the hammer up. The contact motion strongly depends on the physical quantities such as the

Figure 1-7: **Challenge III: Hard to recognize tools due to their thin, texture-less nature and occlusions.** Two examples showing the cases where the tools are hard to recognize due to occlusion of person and background.

mass of the object and the contact forces exerted by the hand, which renders modelling of contacts a very difficult task.

3. **Tool recognition.** As shown in Figure 1-7, the tools we consider in this work, such as hammer, scythe, or spade, are particularly difficult to recognize due to their thin structure, lack of texture, and frequent occlusions by hands and other parts of human body.

1.4 Contributions

The contributions of this thesis are summarized as follows:

1. **Estimating human-object 3D motion and contact forces from a single video.** The first and key contribution of this thesis is an approach, which can jointly estimate the 3D motion of a person-object interaction together with the actuation forces exerted by the person on the manipulated object. This is achieved by modelling contacts and the dynamics of the interactions. It is cast as a large-scale trajectory optimization problem under the constraints of contact force models and the dynamics.

2. **Recognizing contacts from image pixels.** We have developed a method to automatically recognize from the input video the 2D position and timing of contacts between the person and the object or the ground, thereby significantly simplifying the complexity of the optimization.

3. **Estimating object 2D endpoints from image.** We have developed a method to estimate the 2D location of the endpoints of several classes of stick-like objects. The method is built on top of object instance segmentation and is trained from large amounts synthetically generated training data requiring only minimal manual annotations.

4. **Collecting and annotating new datasets for evaluating human 3D motion and forces.** Moreover, we have collected and annotated a new dataset of Internet videos showing people manipulating a variety of tools in unconstrained environments. In addition, we adopted a recent video-MoCap dataset capturing typical parkour actions, post-processed the original data to obtain the ground truth human 3D motion and contact forces to validate our method.

1.5 Outline of the thesis

The remainder of this thesis is structured as follows:

In Chapter 2, we provide a review of the relevant methods for solving problems related to our goal. The problems include estimating human 3D pose and object 6D pose from image pixels from a single viewpoint, optimizing 3D trajectories of a human subject and the manipulated object under physics constraints, manipulating objects with a robot or character animation.

Chapter 3 addresses the problem of extracting 2D measurements from video. This problem breaks down into sub-problems of 2D human pose estimation, contact

recognition (contribution 2) and object 2D endpoint estimation (contribution 3).

In Chapter 4, we solve the problem of estimating human and object 3D motion from the 2D measurements by formulating a trajectory optimization problem under physical constraints (contribution 1). We provide details on how we formulate the optimization problem, discretize it into a numerical problem and solve it.

Chapter 5 presents quantitative and qualitative evaluation of the reconstructed 3D person-object interactions. We first introduce the Handtool dataset and the post-processed LAAS Parkour dataset for evaluating the quality of 3D motion and force estimation (contribution 4). Then we provide a comprehensive study on the proposed method using these two datasets.

Finally, Chapter 6 concludes this thesis by presenting the main obtained results and possible future research directions.

1.6 Publications, software and data

Two research papers have been published during the preparation of this thesis:

- The main parts of Chapter 3 and Chapter 4 were accepted for an oral presentation at the International Conference on Computer Vision and Pattern Recognition 2019 (CVPR'19) [Li et al., 2019]. This paper was also selected among the best paper finalists at the conference.

- An extended version of the CVPR paper has been published at the the International Journal of Computer Vision (IJCV) [Li et al., 2022].

- The new datasets used for evaluating 3D motion and forces are available online at `https://github.com/zongmianli/Handtool-dataset` (the Handtool dataset) and `https://github.com/zongmianli/Parkour-dataset` (the LAAS Parkour dataset).

- All the software developed during this thesis is available online at https://github.com/zongmianli/Estimating-3D-Motion-Forces under an open-source licence.

Chapter 2

Related work

In this chapter, we review recent work in both computer vision and robotics literature that are related to the topic of this thesis. We begin by reviewing methods for estimating human and object pose from single images in Section 2.1 and Section 2.2, respectively. Then we follow in Section 2.3 with a brief review of recent work in learning from instructional videos. Finally, in Section 2.4 we discuss methods for estimating 3D robot-object interactions in optimal control and robotics, typically assuming rigid body models.

2.1 Human pose estimation

Single-view 3D pose estimation aims to recover the 3D joint configuration of the person from the input image. Recent human 3D pose estimators either attempt to build a *direct mapping* from image pixels to the 3D joints of the human body or break down the task into *two stages*: estimating pixel coordinates of the joints in the input image and then lifting the 2D skeleton to 3D.

The existing *direct mapping approaches* either rely on generative models to search the state space for a plausible 3D skeleton that aligns with the image

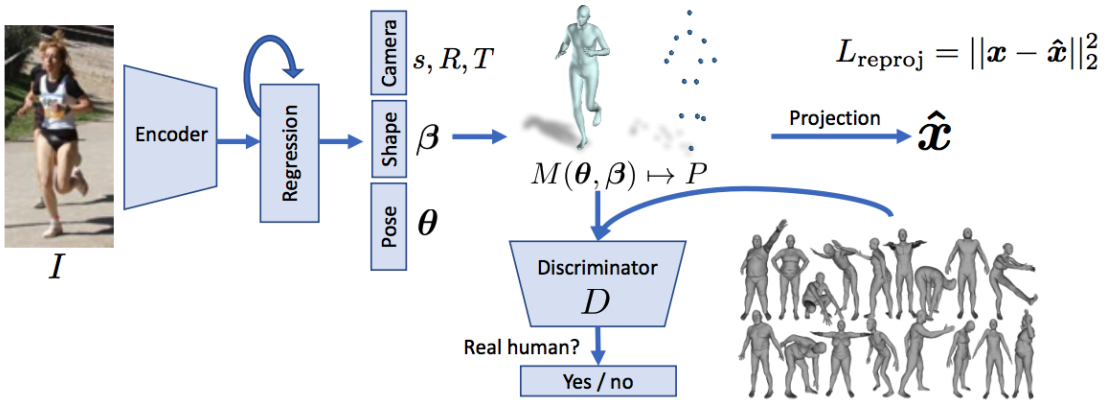


Figure 2-1: **The architecture of Human Mesh Recovery [Kanazawa et al., 2018].** An input image I is passed through a convolutional encoder. This is sent to an iterative 3D regression module that infers the latent 3D representation of the human that minimizes the joint reprojection error. The 3D parameters are also sent to the discriminator D, whose goal is to tell if these parameters come from a real human shape and pose. Figures are taken from Kanazawa et al. [2018].

evidence [Sidenbladh et al., 2000; Gammeter et al., 2008; Gall et al., 2010] or, more recently, extract deep features from images and learn a regressor from the 2D image to the 3D pose [Kanazawa et al., 2018; Moreno-Noguer, 2017; Pavlakos et al., 2017; Tekin et al., 2016]. We illustrate the work of Kanazawa et al. [2018] in Figure 2-1 as a typical example of direct approach. In particular, Kanazawa et al. [2018] train a single-image pose estimator using in-the-wild images that only have ground truth 2D annotations, together with an unpaired dataset of static 3D human shapes and poses through adversarial training. The follow-up work [Kanazawa et al., 2019] shows that the models can be further extended to learn 3D human dynamics from 2D in-the-wild video data collected from Instagram.

On the other hand, *two-stage approaches* have been shown to be very effective for the task of single-view 3D pose estimation [Akhter and Black, 2015; Zhou et al., 2016; Bogo et al., 2016; Chen and Ramanan, 2017], and have achieved competitive results [Martinez et al., 2017; Xiang et al., 2019] on 3D human pose benchmarks

such as Ionescu et al. [2014]. The output can have an impressive level of detail including face deformations and position of individual fingers [Xiang et al., 2019]. The good performance of two-stage approaches is built on top of the recent progress in 2D human pose estimation [Newell et al., 2016, 2017; Insafutdinov et al., 2016]. In particular, the work of Openpose [Cao et al., 2017] has drawn great attention. The overall pipeline of Openpose is illustrated in Figure 2-2.

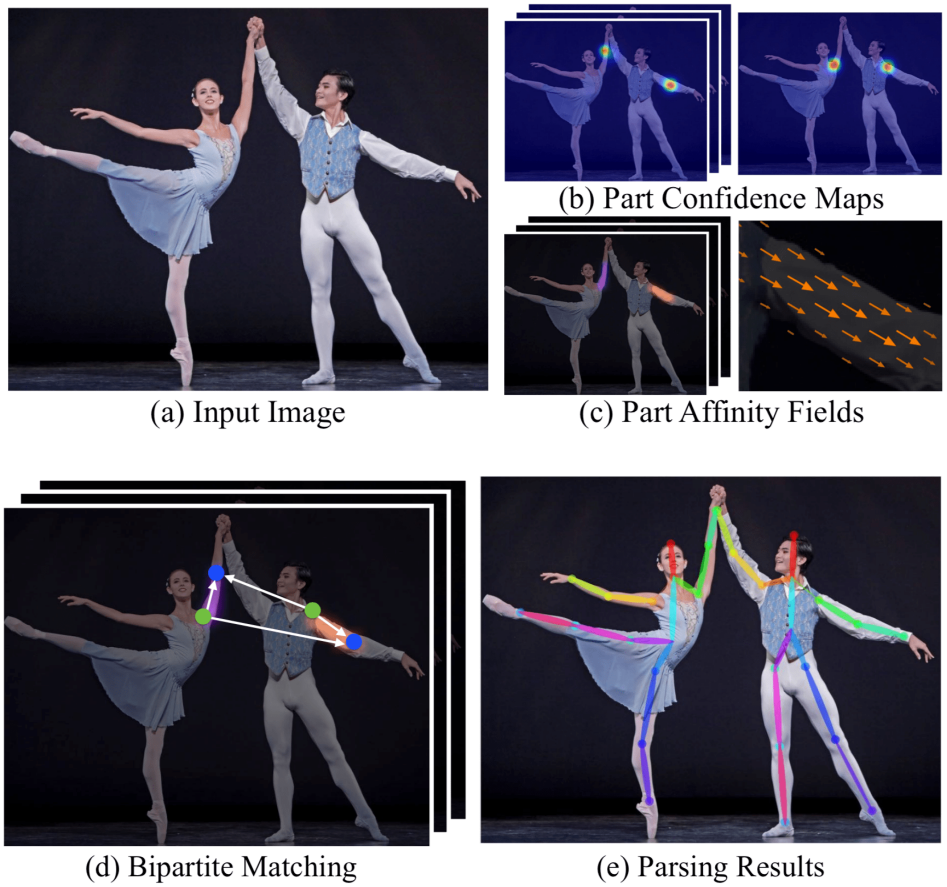


Figure 2-2: **The overall pipeline of Openpose [Cao et al., 2017]**. The method (a) takes the entire image as the input for a CNN to jointly predict (b) confidence maps for body part detection and (c) Part affinity fields (PAFs) for part association. (d) The parsing step performs a set of bipartite matchings to associate body part candidates, which are (e) finally assembled into full body poses for all people in the image. Figures adopted from Cao et al. [2017].

To deal with depth ambiguities, both the direct and the two-stage 3D pose estimators often rely on good pose priors. The priors can be either hand-crafted or learnt from large-scale motion capture data [Zhou et al., 2016; Bogo et al., 2016; Kanazawa et al., 2018; Kocabas et al., 2020]. Others have looked at incorporating physical constraints. Examples include incorporating geometric constraints representing the proximity to the ground plane or collisions between different people [Zanfir et al., 2018], or, closer to our work, modelling the dynamics of the human motion and the contacts with the ground [Rempe et al., 2020; Shimada et al., 2020, 2021]. However, unlike our work, these approaches do not consider explicit physical models for 3D interactions between the person and the manipulated object.

Understanding human-object interactions involves both recognition of actions and modelling of interactions. In action recognition, most existing approaches that model human-object interactions do not consider 3D, instead model interactions and contacts in the 2D image space [Gupta et al., 2009; Delaitre et al., 2011; Yao and Fei-Fei, 2012; Prest et al., 2013]. Recent works in scene understanding [Jiang et al., 2013; Fouhey et al., 2014] consider interactions in 3D but have focused on static scene elements rather than manipulated objects as we do in this work. Tracking 3D poses of people interacting with the environment has been demonstrated for bipedal walking [Brubaker et al., 2007, 2009] or in sports scenarios [Wei and Chai, 2010]. However, these works do not consider interactions with objects. Furthermore, Wei and Chai [2010] requires manual annotation of the input video and does not model the manipulated object. Figure 2-3 illustrates the motion reconstruction approach described in Wei and Chai [2010].

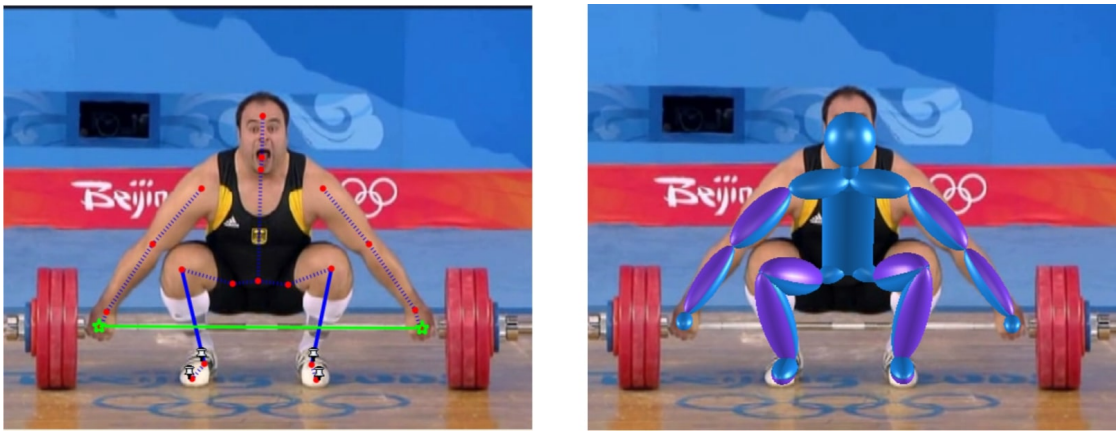


Figure 2-3: **VideoMocap** is a video-based motion modelling technique that is capable to reconstruct physically realistic human motion from monocular video sequences. As shown in the image on the left, this method requires manual annotation of a set of 2D person joints (marked by red dots) and contact points (green dots) to be able to handle the interaction between the human subject and the environment. In addition, this method does not model the manipulated object (barbell in this example) and the ground. Images reproduced from Wei and Chai [2010].

2.2 Object pose estimation

Object 3D pose estimation methods often require depth or RGB-D data as input [Tejani et al., 2014; Doumanoglou et al., 2016; Hinterstoisser et al., 2016], which is restrictive since depth information is not always available (e.g. for outdoor scenes or specular objects), as is the case of our instructional videos. Recent work has also attempted to recover object pose from RGB input only [Brachmann et al., 2016; Rad and Lepetit, 2017; Xiang et al., 2017; Li et al., 2018; Oberweger et al., 2018; Grabner et al., 2018; Rad et al., 2018]. However, we found that the performance of these methods is limited for the stick-like objects we consider in this work. Instead, we recover the 3D pose of the object via localizing and segmenting the object in 2D, and then jointly recovering the 3D trajectory of both the human limbs and the object. As a result, both the object and the human pose help each other to improve their joint 3D trajectory by leveraging the contact constraints.

2.3 Instructional videos

Our work is also related to recent efforts in learning from Internet instructional videos [Malmaud et al., 2015; Alayrac et al., 2016] that aim to segment input videos into clips containing consistent actions, or learn video-language representations [Miech et al., 2019, 2020].

In contrast, we focus on extracting a detailed representation of the object manipulation in the form of a 3D person-object trajectory with contacts and underlying interaction forces.

2.4 Optimal control and robotics

There is also related work on modelling person-object interactions in robotics [Tassa et al., 2012] and computer animation [Boulic et al., 1990]. Similarly to people, humanoid robots interact with the environment by creating and breaking contacts [Herdt et al., 2010], for example, during walking. Typically, generating artificial motion is formulated as an optimal control problem, transcribed into a high-dimensional numerical optimization problem, seeking to minimize an objective function under contact and feasibility constraints [Diehl et al., 2006; Schultz and Mombaur, 2010]. A known difficulty is handling the non-smoothness of the resulting optimization problem introduced by the creation and breaking of contacts [Westervelt et al., 2003]. Due to this difficulty, the sequence of contacts is often computed separately and not treated as a decision variable in the optimizer [Kuffner et al., 2005; Tonneau et al., 2018a]. Recent work has shown that it may be possible to decide both the continuous movement and the contact sequence together, either by implicitly formulating the contact constraints [Posa et al., 2014] or by using invariances to smooth the resulting optimization problem [Mordatch et al., 2012; Winkler et al., 2018]. As a quick example, we illustrate in Figure 2-4 the contact-invariant

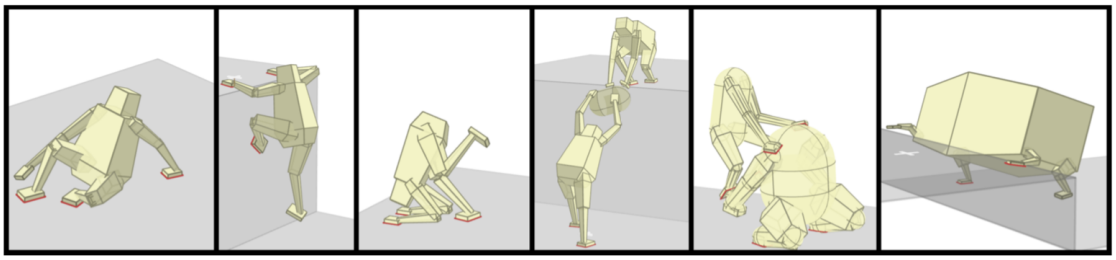


Figure 2-4: **Contact-invariant optimization.** This approach enables simultaneous optimization of contacts and behavior, by augmenting the search space with scalar variables that indicate whether a potential contact should be active in a given phase of the movement. Figure reproduced from Mordatch et al. [2012]

optimization approach proposed by Mordatch et al. [2012].

Learning-based approaches are also emerging in recent years. It is shown that reinforcement learning is capable of learning robust control policies to imitate a broad range of human motion. For example, in the pioneering work of Peng et al. [2018], control policies are learnt directly from RGB video to imitate the action of back-flipping demonstrated by human actors. Figure 2-5 illustrates the approach of Peng et al. [2018] with an Atlas robot in simulation.

In this thesis, we take advantage of rigid-body models introduced in robotics and formulate the problem of estimating 3D person-object interactions from monocular video as an optimal control problem under contact constraints. We overcome the difficulty of contact irregularity by first identifying the contact states from the visual input, and then localizing the contact points in 3D via our trajectory estimator. This allows us to treat multi-contact sequences (like walking) without manually annotating the contact phases.

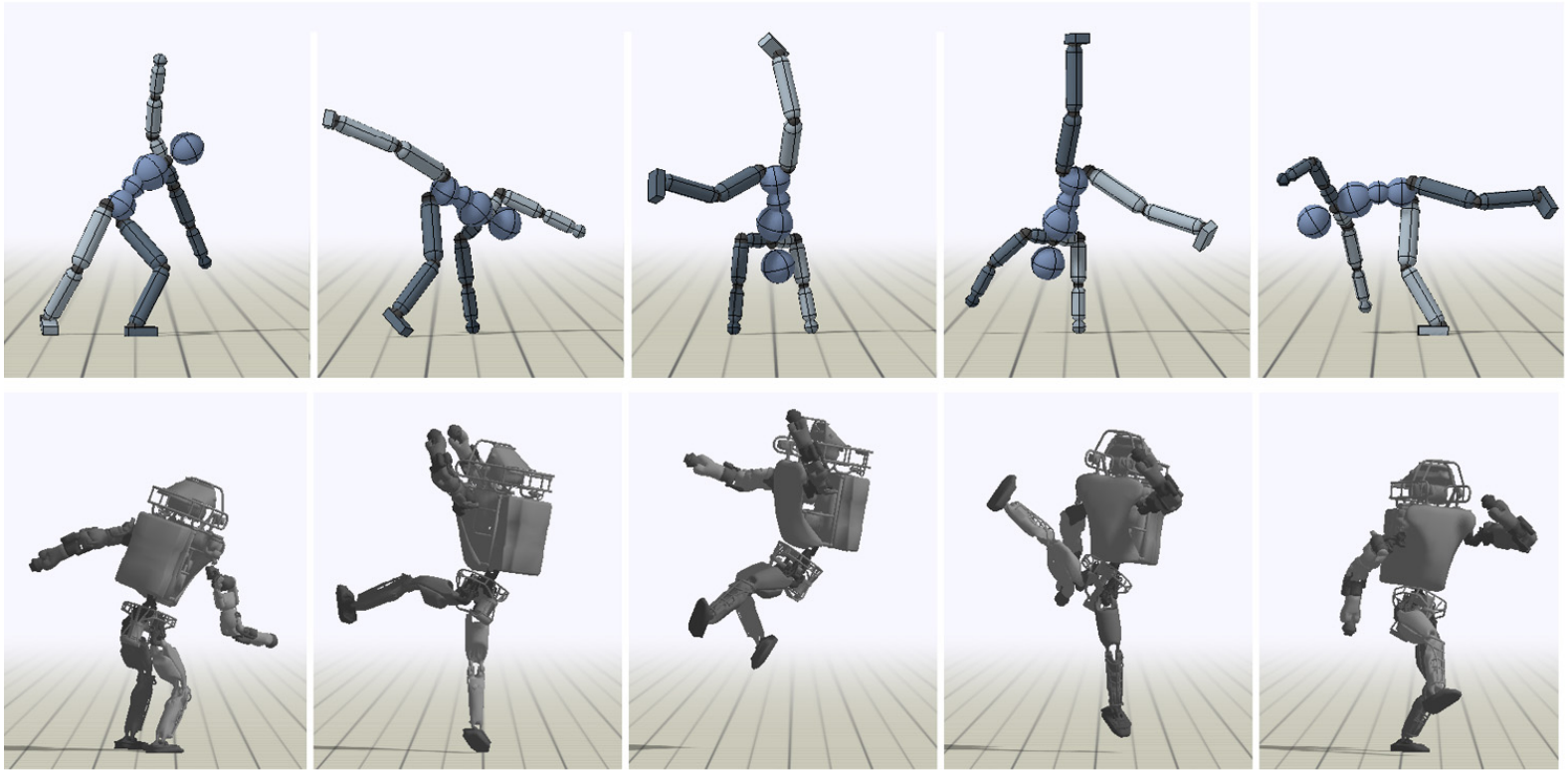


Figure 2-5: **DeepMimic.** This work shows that reinforcement learning is capable of learning robust control policies from RGB video to imitate a broad range of human motion such as back-flipping show in the figure. Figures are taken from Peng et al. [2018].

Chapter 3

Extracting 2D measurements from video

As described in Section 1.5 (Outline of the thesis), our approach has two main stages: the recognition stage and the estimation stage. In this chapter, we describe in detail the first, recognition stage of our approach, in which three types of 2D measurements are extracted from the input video.

In particular, we estimate: (i) the 2D positions of a set of predefined human joints in Section 3.1, (ii) the 2D endpoints positions of a stick-like object in Section 3.2, and (iii) the contact states of the human joints that can potentially touch the object in Section 3.3. Finally, in Section 3.4, we summarize the limitations and the typical failure modes of the proposed approach and discuss possible ways of addressing them.

3.1 Estimating 2D human joints

For estimating 2D human joints from video, we use the recent human 2D pose estimator Openpose [Cao et al., 2017] which achieved excellent performance on

the MPII Multi-Person benchmark [Andriluka et al., 2014]. Taking a pre-trained Openpose model, we do a forward pass on the input video in a frame-by-frame manner. As such we obtain an estimate of the 2D trajectory of human joints observed in the image. Example qualitative results are shown in the column on the right in Figure 3-1.

In particular, we have adapted the original implementation of Openpose (without hands and facial landmarks) to the scenario of object manipulation in instructional videos. The following post-processing steps are appended to the testing module of Openpose: We assume that there is at most one person in the input image or video frame. When multiple human instances are present, only the one detected with the highest confidence score are preserved, while the others are ignored. Due to the heavy occlusion during person-object interaction, the predicted PAFs (part affinity fields as explained eariler in Chapter 2) may not be correct. As a result, some joints (often hands, ankles) may be missing or mis-detected, e.g. associated to another person in the background. To address this problem and deal with such occlusions we have modified the bottom-up parsing step compared to the original implementation.

Figure 3-1: **Estimating 2D human joints from single image.** Example qualitative results. **Left**: Input video frames. **Right**: Frames with estimated Openpose 2D joints.

3.2 Estimating 2D object keypoints from video

The objective is to estimate the 2D position of the manipulated object in each video frame. To achieve this, we build on instance segmentation, computed by Mask R-CNN [He et al., 2017]. We train Mask R-CNN separately for each object class (i.e., barbell, hammer, scythe and spade) and apply it to the corresponding videos. Using the inferred segmentation masks and bounding boxes, we estimate the 2D location of the object endpoints (i.e. its two extremities) in each frame. The resulting 2D endpoint coordinates are used as input to the trajectory optimizer described in detail in Chapter 4. Details are given next.

In order to generate training data for the instance segmentation, we used two different approaches. In the case of barbell, hammer and scythe, we created a 3D model for each object class (i.e. one model for all barbell instances, for example), roughly approximating the shape of the corresponding object instances in the videos, and computed the mask of the model shape in 2D from multiple viewpoints using a perspective camera. For spade, we collected a small number (13) of still images capturing different instances of person-spade manipulation similar to those in the considered videos, and annotated 2D masks of the spade in them. Then we augmented the resulting 2D shape masks to train a separate Mask R-CNN model for each object class. In order to handle the variation of object poses in the videos, we augmented the training set by random 2D geometric transformations (translation, rotation, scale, flip). In addition, to handle the intra-class variation of instance surface appearance as well as changes caused by illumination, we applied domain randomization [Loing et al., 2018; Tobin et al., 2017]: the geometrically transformed 2D mask was filled with a random (foreground) image and pasted on another random (background) image; the random images were taken from the MS COCO dataset [Lin et al., 2014]. Starting with a Mask R-CNN [Abdulla, 2017] model pre-trained on the MS COCO dataset, we trained a separate model for each

object class by fine-tuning the head layers using the corresponding augmented training set.

At test time, we use the segmentation masks and bounding boxes from the trained Mask R-CNN to estimate the 2D coordinates of the object endpoints. In our set-up, the Mask R-CNN is constrained to output no more than one segmented instance per image frame. The endpoints are calculated as the intersection of a line fitted through the segmentation mask (estimate of object's main axis) and the bounding box (estimate of object's extremities). However, we discard the endpoints if the distance of either wrist joint from the line segment between the endpoints is larger than a threshold (incorrect segmentation of the manipulated object). The relative orientation of the object (i.e. which endpoint corresponds to the "head" of the tool and which to its "handle", for example) is determined by the relative proximity of each endpoint to the wrist joints (hammer) or by the relative spatial location of the endpoints in the video frames (barbell, scythe, spade). Figure 3-2 illustrates the output of our object localization and endpoint detection.

Figure 3-2: **Detecting and localizing objects in video frames.** Example qualitative results. **Left**: Input video frame (top to bottom: barbell, hammer, scythe, spade). **Right**: Output object mask (magenta) and object endpoints (yellow and cyan circles, corresponding to the "head" and the "handle" of the tool, respectively, where applicable).

3.3 Recognizing person-object contact points

We wish to recognize and localize contact points between the person and the manipulated object or the ground. This is a challenging task due to the large appearance variation of the contact events in the video.

3.3.1 Proposed approach

However, we demonstrate here that a good performance can be achieved by training a contact recognition CNN module from manually annotated contact data that combine both still images and videos harvested from the Internet. In detail, the contact recognizer operates on the 2D human joints predicted by Openpose (Section 3.1). As shown in Figure 3-3, given 2D joints at video frame i, we crop fixed-size image patches around a set of joints of interest, which may be in contact with an object or ground. Based on the type of human joint, we feed each image patch to the corresponding CNN to predict whether the joint appearing in the patch is in contact or not. The output of the contact recognizer is a sequence δ_{ji} encoding the contact states of human joint j at video frame i, i.e. $\delta_{ji} = 1$ if joint j is in contact at frame i and zero otherwise. Note that δ_{ji} is the discretized version of the contact state trajectory δ_j presented previously.

Our contact recognition CNNs are built by replacing the last layer of an ImageNet pre-trained Resnet model [He et al., 2016] with a fully connected layer that has a binary output. We have trained separate models for five types of joints: hands, knees, foot soles, toes, and neck. To construct the training data, we collect still images of people manipulating tools using Google image search. We also collect short video clips of people manipulating tools from Youtube in order to also have non-contact examples. We run Openpose pose estimator on this data, crop patches around the 2D joints, and annotate the resulting dataset with contact states.

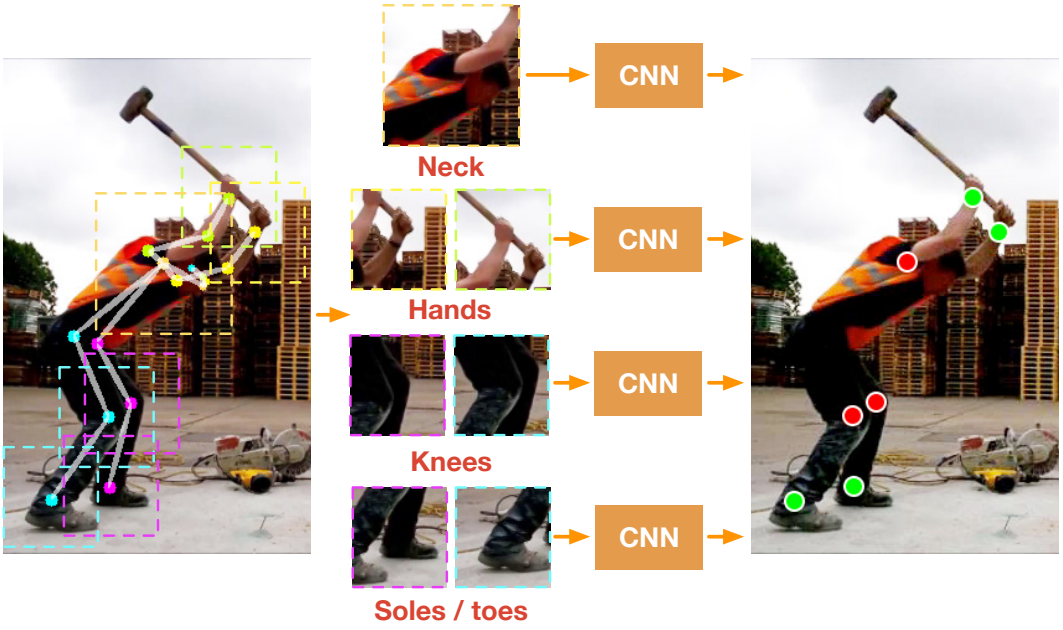


Figure 3-3: The main contact recognition steps. Given estimated 2D human joints, we crop image patches around a set of joints of interest, which includes neck, hands, knees, foot soles and toes. Based on the type of human joint, we feed each image patch to the corresponding CNN to predict whether the joint appearing in the patch is in contact (shown in green on the right) or not (shown in red) with the environment.

3.3.2 Evaluation

In this section, we evaluate the quality of our contact recognizers that is described previously.

To form the test set, we annotate contact states in the entire Handtool dataset and a subset of the Parkour dataset obtained by sampling every 5-th frame. Following the same annotation process as done for training, we have cropped image patches around individual human joints in the test set. This results in a separate test set for each of the five joint types: hand, sole, toes, neck and knee. The neck and the knee test sets include only patches from the Handtool dataset as the Parkour dataset does not consider these types of contacts.

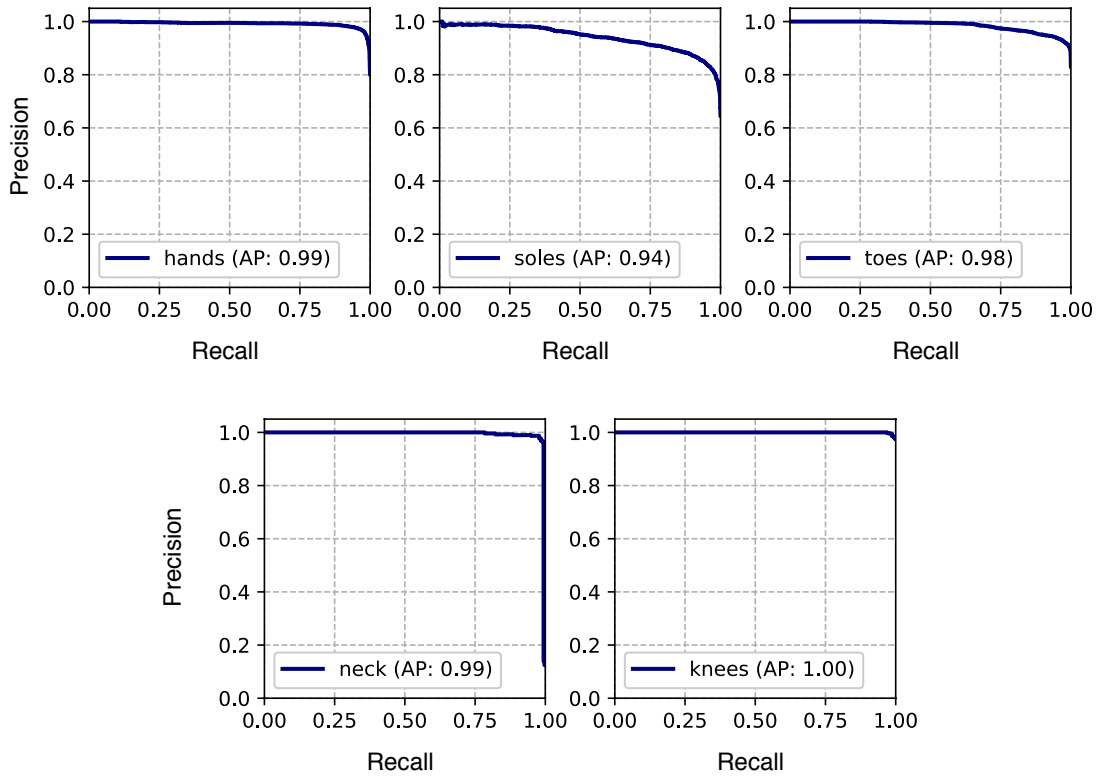


Figure 3-4: Precision-recall curves of our trained models for recognizing the hand-, sole-, toes-, neck-, and knee-contact state.

We evaluate each contact recognizer using a precision-recall curve on its corresponding test set. The positive class means the joint is "in contact". The evaluation results are shown in Fig. 3-4. Each precision-recall curve is also summarized using average precision (AP). The results demonstrate good quality of our contact recognition models despite the appearance variation present in both the Handtool and Parkour datasets.

Finally, we present some qualitative examples of our contact recognition CNNs, tested on the Handtool dataset. The results are illustrated in Figure 3-5.

Figure 3-5: **Recognizing contact states.** Example qualitative results on the Handtool dataset. **Left**: Input video frames. **Right**: Output contact states (green in contact versus red not in contact).

3.4 Discussion of limitations and failure modes

While the proposed methods achieve good performance in extracting 2D measurements, they may still encounter difficulties in some challenging situations. In this section, we summarize the typical failure modes of the three proposed components in the recognition stage. We analyze these problems and discuss possible ways of alleviating or overcoming them.

Limitations in estimating human 2D joint positions. Our human 2D pose estimation module has inherited some problems from Openpose [Cao et al., 2017], based on which it is developed. As mentioned earlier, Openpose is a multi-person 2D pose estimator which will return multiple human subjects when multiple people are present in the input image. Assuming the scenario of a single person manipulating a tool, our method will only keep the main human subject in the foreground while ignoring the others in the background. This is done by ranking the confidence scores of all detected subjects and keeping the one with the highest score. An example is shown in Figure 3-6a, where our pose estimation module successfully focuses on the subject of interest (SoI in short) in grey lifting the barbell, while all the other people are ignored. However, this strategy may fail to focus on the SoI if the person's confidence score is low while another high-confident human instance is detected. Figure 3-6b illustrates such a failure example from the same input video. In this particular case, the SoI's body and facial features are hidden when they grip the barbell bar, and therefore the person in the back is detected instead.

Related to this is the problem of partial occlusion. To address the problem of partial occlusion and to increase the number of detected joints, we have redesigned the strategy of the bottom-up parsing step in the original implementation of Openpose. However, this may lead to the problem of body joint assignment when multiple people are present. As shown in Figure 3-6c, the upper body joints of the

SoI are not detected due to occlusion, and our software has detected and incorrectly assigned the wrong joints to the SoI.

Limitations of estimating object 2D keypoints. Videos capturing object manipulation scenes usually include heavy occlusions between human limbs and the manipulated tool. Our instance segmentation network may fail to detect the object when it is occluded by the person's body at certain frames in the input video. In this case, the estimated 2D endpoint sequences include incorrect observations at those frames with heavy occlusions. This is shown in the first example in Figure 3-7 where the hammer's handle tip is completely occluded by the person's left hand (see the input image on the left). This leads to an incorrect segmentation of the hammer handle followed by an incorrect estimate of the 2D location of the handle tip (marked by a cyan circle in the output visualization on the right).

Another typical failure mode is the incorrect labelling of the head and the handle tips of the stick-like tool which we are considering. As shown in the second row of Figure 3-7, the object segmentation is correct but the handle tip held by the person's left hand is labelled as a hammer head.

Finally, the 2D keypoint estimation module presented in this chapter can only deal with stick-like objects. More objects, for example in box-like shapes, can no longer be described by only two keypoints.

Limitations of recognizing contacts. Our contact recognizer works well in the type of instructional videos and environments we consider, but may struggle to generalize to new environments.

Currently, the proposed contact recognition network only handles three types of contacts: the contact between a person hand and a stick-like handle, the contact between foot/knee and the ground and the contact between neck and the barbell bar. The datasets used for training only contain these three types of contacts.

More complicated contacts such as self-body contacts of a person clapping his/her hands are considered to be less important and are ignored, although they can also affect the dynamics of the person's movement.

The size of the training data is limited due to the difficulty of collecting object manipulating videos and annotating the contacts manually. Therefore, our contact recognition network achieves better performance on videos that are captured in similar environments as those contained in the training videos. In addition, the training data does not contain appearance variations of human limbs, e.g. hand size, skin color, etc.

(a) Subject of interest (SoI) detection in success.

(b) Failure due to multiple human subjects. This is because the SoI's confidence score is lower than the score of the person in the back.

(c) Failure due to incorrect body part assignment. The upper body joints of the SoI are occluded and hence undetected.

Figure 3-6: **Typical failure modes in estimating human 2D joint positions.** The example frames are from the same input video with multiple human instances in the background.

Figure 3-7: **Typical failure modes in object 2D endpoint estimation.** The yellow and cyan circles correspond to the "head" and the "handle" of the tool, respectively. **Top**: An example frame showing incorrect object mask estimation, which leads to incorrect localization of the handle (cyan circle). **Bottom**: An example with correct position of predicted endpoints but flipped head and handle of the tool.

Chapter 4

Estimating 3D motion and forces using 2D measurements

In this chapter, we describe the second, estimation, stage of our approach. In particular, we develop a model for reliably estimating the 3D person-object motion and contact forces from the 2D measurements obtained in the recognition stage described in Chapter 3. Contrary to the deep learning approaches used in Chapter 3, in this chapter we rely on numerical optimization to jointly estimate the person-object 3D motion and forces under physics constraints. In particular, we formulate a large-scale trajectory optimization problem to model the dynamics of 3D movement of the person, the manipulated object, and their interaction with created and broken physical contacts.

4.1 Parametric human and object models

Human model. We model the human body as a multi-body system consisting of a set of rotating joints and rigid links connecting them. We adopt the joint definition of the SMPL model [Loper et al., 2015] and approximate the human

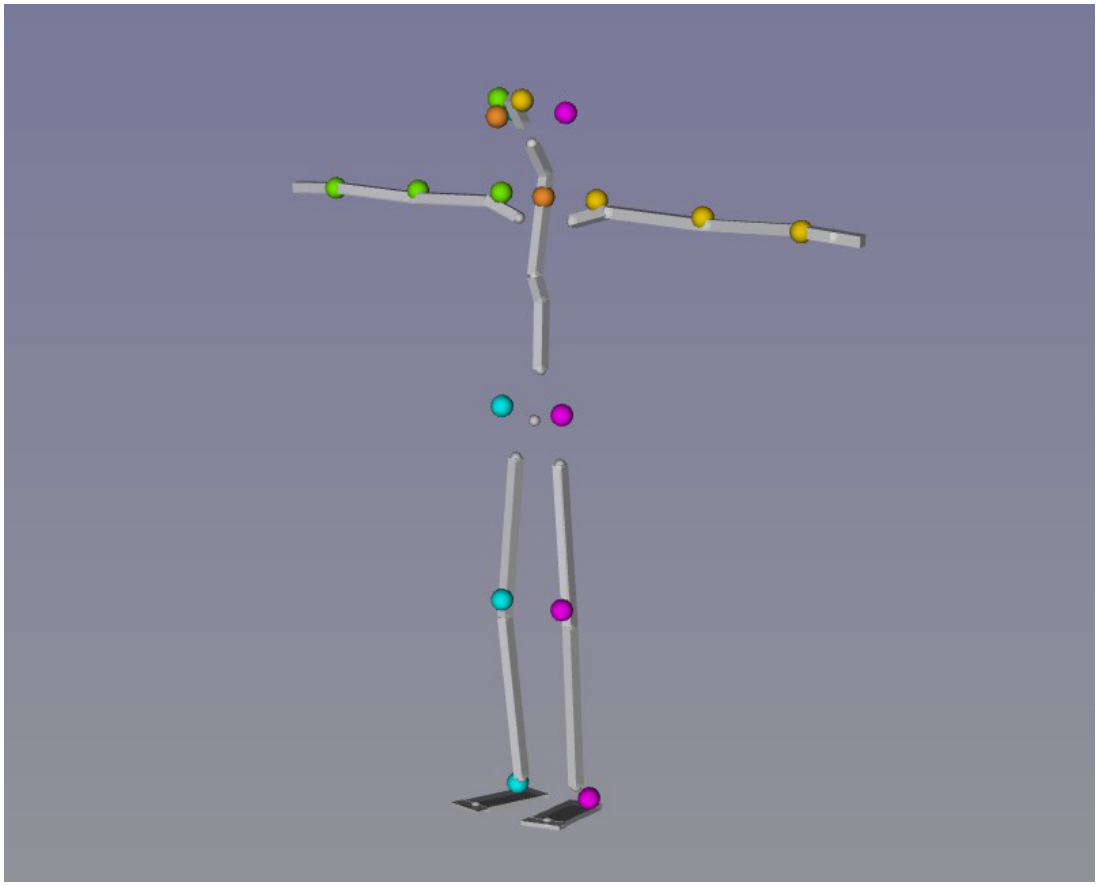


Figure 4-1: An illustration of our parametric human model in the reference posture. The skeleton in grey consists of one free-floating basis joint corresponding to the pelvis, and 23 spherical joints representing the major moving joints of human body. The colored spheres are 18 virtual markers that correspond to 18 OpenPose joints. Each marker is associated to a semantic joint in our model.

skeleton as a kinematic tree with 24 joints: one free-floating joint and 23 spherical joints. Figure 4-1 illustrates our human model in a canonical pose. A free-floating joint consists of a 3-dof translation in $\mathbb{R}^3$ and a 3-dof rotation in $SO(3)$; we model the pelvis by a free-floating joint to describe the person's body orientation and translation in the world coordinate frame. A spherical joint is a 3-dof rotation; it represents the relative rotation between two connected links in our model. In practice, we use unit quaternions to represent 3D rotations and axis-angles to

describe angular velocities. As a result, the configuration vector of our human model q^{h} is a concatenation of the configuration vectors of the 23 spherical joints (dimension 4) and the free-floating pelvis joint (dimension 7), hence of dimension 99. The corresponding human joint velocity $\dot{q}^{\mathrm{h}}$ is of dimension $23 \times 3 + 6 = 75$ (by replacing the quaternions with axis-angles). For simplicity, in the main paper we do not distinguish this difference in dimension and consider both q^{h} and $\dot{q}^{\mathrm{h}}$ to be represented using axis-angles, hence of the same dimension $n_q^{\mathrm{h}} = 75$. In addition, based on these 24 joints, we define 18 "virtual markers" (shown as colored spheres in Figure 4-1) that represent the 18 OpenPose joints. These markers are used instead of the 24 joints to compute the re-projection errors with respect to the OpenPose 2D detections.

Object models. All four objects, namely barbell, hammer, scythe and spade, are modeled as a non-deformable rigid line stick. The configuration q^{o} represents the 6-dof displacement of the stick handle, as illustrated in Figure 4-2. In practice, q^{o} is a 7-dimensional vector containing the 3D translation and 4D quaternion rotation of the free-floating handle end. The object joint velocity $\dot{q}^{\mathrm{o}}$ is of dimension 6 (by replacing the quaternion with an axis-angle). The handtools that we are modelling have the stick handle as the contact area. We ignore the handle's thickness and represent the contact area using the line segment between the two endpoints of the handle. Depending on the number of human joints in contact with the object, we associate the same number of contact points to the object's local coordinate frame. These contact points can be located at any point along the feasible contact area. In practice, all object contact points together with the endpoint corresponding to the head of the handtool are implemented as "virtual" prismatic joints of dimension 1.

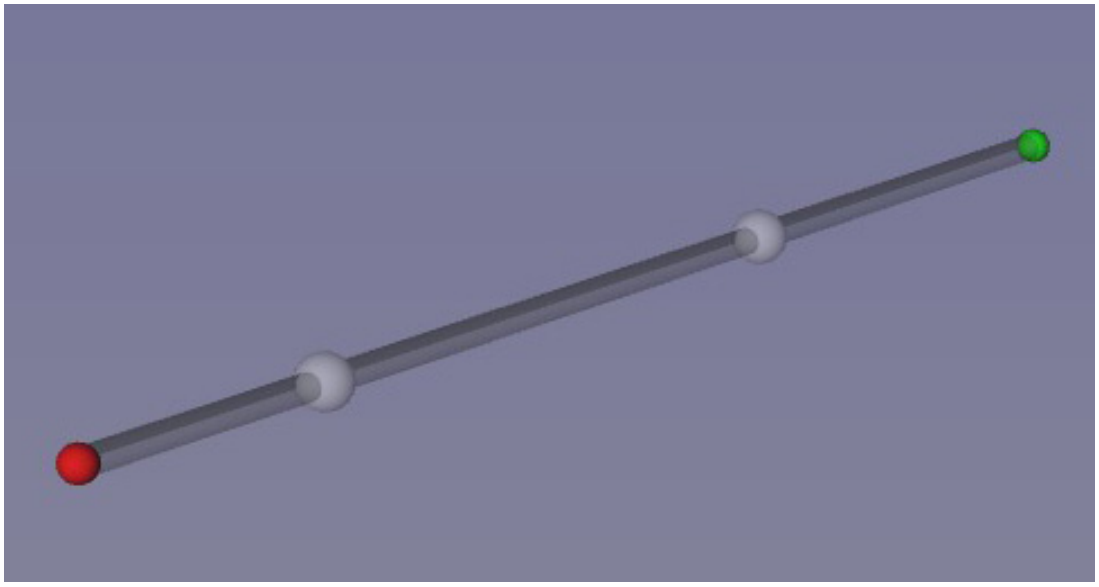


Figure 4-2: All the four types of handtools are represented by a single parametric object model, as shown in the image. The object model consists of 1 free-floating basis joint corresponding to the handle end point (red sphere), 1 prismatic joint corresponding to the head of the tool (green sphere), and several prismatic joints corresponding to the location of the contact points (grey translucent spheres in the middle). The contact points should lie on the feasible contact area (grey stick) formed by the two endpoints.

4.2 Assumptions

The proposed problem is difficult, yet feasible to solve under a number of reasonable assumptions on the physical properties of the person, the manipulated object and the scene.

First of all, we assume that there is at most one person that appears in the input video. We adopt the mass properties of the full-body anatomical human model described in [Maldonado, 2018]. This model captures the body weight statistics of an average human adult. Our approach applies the same body mass distribution to any input video despite the variation of shape of the person in the video.

If there is an object manipulated by the person, we assume that the object is rigid, non-articulated and has a stick-like shape. We apply a single object mass

distribution to any input video with the same type of object. For example, we assume that all sledgehammers share the same head weight. Our method can also handle input videos without the manipulated object. In this case, we only model contacts between the person and the ground, with no object model and the corresponding 2D inputs.

We further assume the camera is static with canonical (or known) intrinsic parameters. Most body joints, especially the ones that may interact with the environment (e.g. hands, feet, knees, etc) should be visible at least in a short period of time in the input video. We assume that the gravity is perpendicular to the ground plane, but the model can be tuned to fit other cases such as a sloping ground.

In the subsequent sections, we will include an object model in our formulation, but as discussed above the object is not necessary for the model to be applied.

4.3 Proposed approach

Let T be the duration of the input video clip depicting person manipulating an object. We encode the 3D poses of the human and the object, including joint translations and rotations, in the configuration vectors q^{h} and q^{o}, for the human and the object respectively. We define a constant set of K contact points between the human body and the object (or the ground plane). Each contact point corresponds to a human segment, and is activated whenever that human segment is recognized as in contact. At each contact point, we define a contact force f_k, whose value is non-zero whenever the contact point k is active. The state of the complete dynamical system is then obtained by concatenating the human and the object joint configurations q and velocities $\dot{q}$ as $x \coloneqq \left(q^{\mathrm{h}}, q^{\mathrm{o}}, \dot{q}^{\mathrm{h}}, \dot{q}^{\mathrm{o}}\right)$. Let $\tau_{\mathrm{m}}^{\mathrm{h}}$ be the joint torque vector describing the actuation by human muscles. This is a $n_q - 6$ dimensional

vector where n_q is the dimension of the human body configuration vector. We define the control variable u as the combination of the joint torque vector together with the contact forces at the K contact point, $u \coloneqq \left(\tau_{\mathrm{m}}^{\mathrm{h}}, f_k, k = 1, ..., K\right)$. To deal with sliding contacts, we further define a contact state c that consists of the relative positions of all the contact points with respect to the object (or ground) in the 3D space.

Our goal is two-fold. We wish to (i) estimate smooth and consistent human-object and contact trajectories $\underline{x}$ and $\underline{c}$, while (ii) recovering the control $\underline{u}$ which gives rise to the observed motion[2]. This is achieved by jointly optimizing the 3D trajectory $\underline{x}$, contacts $\underline{c}$, and control $\underline{u}$ given the measurements (2D positions of human joints and object endpoints together with contact states of human joints) obtained from the input video. The intuition is that the human and the object's 3D poses should match their respective projections in the image while their 3D motion is linked together by the recognized contact points and the corresponding contact forces. In detail, we formulate person-object interaction estimation as an optimal estimation problem with contact and dynamics constraints:

$$\underset{\underline{x},\underline{u},\underline{c}}{\text{minimize}} \quad \sum_{e\in\{\mathrm{h,o}\}} \int_0^T l^e\left(x, u, c\right) \mathrm{d}t, \tag{4.1}$$

$$\text{subject to} \quad \kappa(x, c) = 0 \quad \text{(contact motion model)}, \tag{4.2}$$

$$\dot{x} = f\left(x, c, u\right) \quad \text{(full-body dynamics)}, \tag{4.3}$$

$$u \in \mathcal{U} \quad \text{(force model)}, \tag{4.4}$$

where e denotes either 'h' (human) or 'o' (object), and the constraints (4.2)-(4.4) must hold for all $t \in [0, T]$. The loss function l^e is a weighted sum of multiple costs capturing (i) the data term measuring simultaneously the consistency between the observed and re-projected 2D joint and object endpoint positions and the

[2]In this paper, trajectories are denoted as underlined variables, e.g. $\underline{x}, \underline{u}$ or $\underline{c}$.

discrepancy of the estimated 3D joint positions with respect to some reference positions, (ii) the prior on the human 3D poses, (iii) the physical plausibility of the motion and (iv) the temporal smoothness of the estimated trajectory. Next, we describe these cost terms as well as the insights leading to their design choices. For simplicity, we ignore the superscript e when introducing a cost term that exists for both the human l^{h} and the object l^{o} component of the loss. We describe the individual terms using continuous time notation as used in the overall problem formulation (4.1). A discrete version of the problem as well as the optimization and implementation details are relegated to Section 4.4.

4.3.1 Data term: enforcing 2D and 3D consistency

Given the 2D locations of human joints and object endpoints predicted from image, we wish to optimize a 3D pose trajectory that consolidates these 2D measurements. This is done by minimizing the re-projection error of the estimated 3D human joints and 3D object endpoints with respect to the 2D measurements obtained in each video frame. In detail, let $j = 1, ..., N$ be human joints or object endpoints and p_j^{2D} their 2D position observed in the image. We minimize the 2D consistency loss l_{2D}:

$$l_{\text{2D}} = \sum_j \rho \left(p_j^{\text{2D}} - P_{\text{cam}}(p_j(q)) \right), \tag{4.5}$$

where P_{cam} is the camera projection matrix and p_j the 3D position of joint or object endpoint j induced by the person-object configuration vector q. To deal with outliers, we use the robust Huber loss, denoted by ρ.

In addition, we employ a direct 3D consistency loss if a reference 3D pose trajectory is available:

$$l_{\text{3D}} = \sum_j \rho \left(p_j^{\text{3D}} - p_j(q) \right), \tag{4.6}$$

where p_j^{3D} denotes the reference 3D position of joint j. In our case, the reference human 3D poses are computed using the HMR estimator [Kanazawa et al., 2018]. But it is possible to use other pose estimators instead.

In practice, we find that minimizing a weighted sum of the 2D and 3D consistency losses achieves good performance. The data term is finally expressed as:

$$l_{\text{data}} = w_{2D} l_{2D} + w_{3D} l_{3D}, \tag{4.7}$$

where w_{2D} and w_{2D} are non-negative scalars.

4.3.2 Prior on 3D human poses

A single 2D skeleton can be a projection of multiple 3D poses, many of which are unnatural or impossible exceeding the human joint limits. To resolve this, we incorporate into the human loss function l^{h} a pose prior similar to Bogo et al. [2016]. The pose prior is obtained by fitting the SMPL human model [Loper et al., 2015] to the CMU MoCap dataset using MoSh [Loper et al., 2014] and fitting a Gaussian Mixture Model (GMM) to the resulting SMPL 3D poses. We map our human configuration vector q^{h} to a SMPL pose vector θ and compute the likelihood under the pre-trained GMM

$$l_{\text{pose}}^{\text{h}} = -\log\left(p(q^{\text{h}}; \text{GMM})\right). \tag{4.8}$$

During optimization, $l_{\text{pose}}^{\text{h}}$ is minimized in order to favor more plausible human poses against rare or impossible ones.

4.3.3 Physical plausibility of the motion

Human-object interactions involve contacts coupled with interaction forces, which are not included in the data-driven cost terms (4.7) and (4.8). Modeling contacts and physics is thus important to reconstruct object manipulation actions from the input video. Next, we outline models for describing the motion of the contacts and the forces at the contact points. Finally, the contact motions and forces, together with the system state $\underline{x}$, are linked by the laws of mechanics via the dynamics equations, which constrain the estimated person-object interaction. This full body dynamics constraint is detailed at the end of this subsection.

Contact motions. In the recognition stage, our contact recognizer predicts, given a human joint (for example, left hand, denoted by j), a sequence of contact states $\delta_j : t \longrightarrow \{1, 0\}$. Similarly to Carpentier and Mansard [2018b], we call a *contact phase* any time segment in which j is in contact, i.e., $\delta_j = 1$. Our key idea is that the 3D distance between human joint j and the active contact point on the object (denoted by k) should remain zero during a contact phase:

$$\left\| p_j^{\mathrm{h}}(q^{\mathrm{h}}) - p_k^{\mathrm{c}}(x, c) \right\| = 0 \quad \text{(point contact)}, \tag{4.9}$$

where p_j^{h} and p_k^{c} are the 3D positions of joint j and object contact point k, respectively. Note that position of the object contact point $p_k^{\mathrm{c}}(x, c)$ depends on the state vector x describing the human-object configuration and the relative position c of the contact along the object. The position of contact p_k^{c} is subject to a feasible range denoted by $\mathcal{C}$. For stick-like objects such as hammer, $\mathcal{C}$ is approximately the 3D line segment representing the handle. For the ground, the feasible range $\mathcal{C}$ is a 3D plane. In practice, we implement $p_k^{\mathrm{c}} \in \mathcal{C}$ by putting a constraint on the trajectory of relative contact positions $\underline{c}$.

Equation (4.9) applies to most common cases where the contact area can be modeled as a point. Examples include the hand-handle contact and the knee-ground contact. To model the *planar contact* between the human sole and ground, we approximate each sole surface as a planar polygon with four vertices, and apply the point contact model at each vertex. In our human model, each sole is attached to its parent ankle joint, and therefore the four vertex contact points of the sole are active when $\delta_{\text{ankle}} = 1$.

The resulting overall contact motion function κ in problem (4.1) is obtained by unifying the point and the planar contact models:

$$\kappa(x, c) = \sum_{j} \sum_{k \in \phi(j)} \delta_j \left\| T^{(kj)} \left(p_j^{\text{h}}(q^{\text{h}}) \right) - p_k^{\text{c}}(x, c) \right\|, \tag{4.10}$$

where the external sum is over all human joints. The internal sum is over the set of active object contact points mapped to their corresponding human joint j by mapping $\phi(j)$. The mapping $T^{(kj)}$ translates the position of an ankle joint j to its corresponding k-th sole vertex; it is an identity mapping for non-ankle joints.

Contact forces. During a contact phase of the human joint j, the environment exerts a contact force f_k on each of the active contact points in $\phi(j)$. f_k is always expressed in contact point k's local coordinate frame. We distinguish two types of contact forces: (i) 6D spatial forces exerted by objects and (ii) 3D linear forces due to ground friction. In the case of object contact, f_k is an unconstrained 6D spatial force with 3D linear force and 3D moment. In the case of ground friction, f_k is constrained to lie inside a 3D friction cone $\mathcal{K}^3$ (also known as the quadratic Lorentz "ice-cream" cone [Carpentier and Mansard, 2018b]) characterized by a positive friction coefficient μ. In practice, we approximate $\mathcal{K}^3$ by a 3D pyramid spanned by a basis of $N = 4$ generators, which allows us to represent f_k as the

convex combination

$$f_k = \sum_{n=1}^{N} \lambda_{kn} g_n^{(3)}, \tag{4.11}$$

where $\lambda_{kn} \geq 0$ and $g_n^{(3)}$ with $n = 1, 2, 3, 4$ are the 3D generators of the contact force. We sum the contact forces induced by the four sole-ground contact points and express a unified contact force in the ankle's frame:

$$f_j = \sum_{k=1}^{4} \begin{pmatrix} f_k \\ p_k \times f_k \end{pmatrix} = \sum_{k=1}^{4} \sum_{n=1}^{N} \lambda_{jkn} g_{kn}^{(6)}, \tag{4.12}$$

where p_k is the position of contact point k expressed in joint j's (left/right ankle) frame, $\times$ is the cross product operator, $\lambda_{jkn} \geq 0$, and $g_{kn}^{(6)}$ are the 6D generators of f_j. Additional details including the expressions of $g_n^{(3)}$ and $g_{kn}^{(6)}$ can be found at the end of the chapter in Section 4.5

Full body dynamics. The full-body movement of the person and the manipulated object is described by the Lagrange dynamics equation:

$$M(q)\ddot{q} + b(q, \dot{q}) = g(q) + \tau, \tag{4.13}$$

where M is the generalized mass matrix, b covers the centrifugal and Coriolis effects, g is the generalized gravity vector and τ represents the joint torque contributions. $\dot{q}$ and $\ddot{q}$ are the joint velocities and joint accelerations, respectively. Note that (4.13) is a unified equation which applies to both human and object dynamics, hence we drop the superscript e here. Only the expression of the joint torque τ differs between the human and the object and we give the two expressions next.

For human, it is the sum of two contributions: the first one corresponds to the internal joint torques (exerted by the muscles for instance) and the second one

comes from the contact forces:

$$\tau^{\mathrm{h}} = \begin{pmatrix} \mathbf{0}_6 \\ \tau_{\mathrm{m}}^{\mathrm{h}} \end{pmatrix} + \sum_{k=1}^{K} \left(J_k^{\mathrm{h}}\right)^T f_k, \tag{4.14}$$

where $\tau_{\mathrm{m}}^{\mathrm{h}}$ is the human joint torque exerted by muscles, f_k is the contact force at contact point k and J_k^{h} is the Jacobian mapping human joint velocities $\dot{q}^{\mathrm{h}}$ to the Cartesian velocity of contact point k expressed in k's local frame. Let n_q^{h} denote the dimension of q^{h}, $\dot{q}^{\mathrm{h}}$ and $\ddot{q}^{\mathrm{h}}$, then $\tau_{\mathrm{m}}^{\mathrm{h}}$ and J_k^{h} are of dimension $n_q^h - 6$ and $3 \times n_q^h$, respectively. We model the human body and the object as two free-floating base systems. In the case of human body, the six first entries in the configuration vector q correspond to the 6D pose of the free-floating base (translation + orientation), which is not actuated by any internal actuators such as human muscles. This constraint is taken into consideration by adding the zeros in Eq. (4.14).

In the case of the manipulated object, there is no actuation other than the contact forces exerted by the human. Therefore, the object torque is expressed as

$$\tau^{\mathrm{o}} = - \sum_{\text{object contact } k} \left(J_k^{\mathrm{o}}\right)^T f_k, \tag{4.15}$$

where the sum is over the object contact points, f_k is the contact force, and J_k^{o} denotes the object Jacobian, which maps from the object joint velocities $\dot{q}^{\mathrm{o}}$ to the Cartesian velocity of the object contact point k expressed in k's local frame. J_k^{o} is a $3 \times n_q^{\mathrm{o}}$ matrix where n_q^{o} is the dimension of object configuration vectors q^{o}, $\dot{q}^{\mathrm{o}}$ and $\ddot{q}^{\mathrm{o}}$.

We concatenate the dynamics equations of both human and object to form the overall dynamics in Eq. (4.3) in problem (4.1), and include a *muscle torque* term $l_{\mathrm{torque}}^{\mathrm{h}} = \|\tau_{\mathrm{m}}^{\mathrm{h}}\|^2$ in the overall cost. Minimizing the muscle torque acts as a regularization over the energy consumption of the human body.

4.3.4 Enforcing the trajectory smoothness

Regularizing human and object motion. Taking advantage of the temporal continuity of video, we minimize the sum of squared 3D joint velocities and accelerations to improve the smoothness of the person and object motion and to remove incorrect 2D poses. We include the following *motion smoothing* term to the human and object loss in (4.1):

$$l_{\text{smooth}} = \sum_j \left(\|\nu_j(q, \dot{q})\|^2 + \|\alpha_j(q, \dot{q}, \ddot{q})\|^2 \right), \tag{4.16}$$

where ν_j and α_j are the spatial velocity and the spatial acceleration[3] of joint j, respectively. In the case of object, j represents an endpoint on the object. By minimizing l_{smooth}, both the linear and angular movements of each joint/endpoint are smoothed simultaneously.

Regularizing contact motion and forces. In addition to regularizing the motion of the joints, we also regularize the contact states and control by minimizing the velocity of the contact points, the temporal variation of the contact forces and the magnitude of the contact forces. The goal is achieved by the minimizing the temporal variation of the contact positions and the contact forces. In addition, we regularize the magnitude of the contact forces according to their locations: we use higher weight penalties for hand contact forces and smaller weight penalties for ground contact forces. As such to favor bigger ground contact forces when both hand and ground contacts are recognized. The is implemented by including the

[3]Spatial velocities (accelerations) are minimal and unified representations of linear and angular velocities (accelerations) of a rigid body [Featherstone, 2008]. They are of dimension 6.

following *contact smoothing* term in the cost function in problem (4.1):

$$l^{\mathrm{c}}_{\mathrm{smooth}} = \sum_{j} \sum_{k \in \phi(j)} \left(\omega_k \|\dot{c}_k\|^2 + \gamma_k \|\dot{f}_k\|^2 + \zeta_k \|f_k\|^2 \right), \tag{4.17}$$

where $\dot{c}_k$ and $\dot{f}_k$ represent, respectively, the temporal variation of the position and the contact force at contact point k. f_k is the contact force at contact point k. ω_k, γ_k and ζ_k are scalar weights of the regularization terms. Note that some contact points, for example the four contact points of the human sole during the sole-ground contact, should remain fixed with respect to the object or the ground during the contact phase. To tackle this, we use a higher ω_k for sole contact points to prevent the foot sole form sliding. We also found important to use higher ζ_k for hand contact forces and smaller ζ_k for ground contact forces to favor larger ground contact forces when both hand and ground contacts are recognized.

4.4 Optimization

4.4.1 Conversion to a numerical optimization problem

We convert the continuous problem (4.1) into a discrete nonlinear optimization problem using the collocation approach [Biegler, 2010]. All trajectories are discretized and the constraints (4.2), (4.3), (4.4) are only enforced on the "collocation" nodes of a time grid matching the discrete sequence of video frames. The optimization variables are the sequence of human and object poses $[x_0 ... x_T]$, torque and force controls $[u_1 ... u_T]$, contact locations $[c_0 ... c_T]$, and the ground plane. We replace the integral in the objective function by a sum over video frames, and rewrite the cost and constraint terms which include derivatives of the state (e.g. joint accelerations) by approximating the derivatives with the backward finite difference scheme (e.g. $a_t := (v_t - v_{t-1})/\Delta t$, with Δt the duration between two video frames).

The resulting problem is nonlinear, constrained and sparse (due to the sequential structure of trajectory optimization).

The problem after discretization becomes a large, sparse and non-linear optimization problem. This is because the discretized objective function becomes a sum of terms that each depend on one time sample (denoted by i) and a subset of the variables $[x_i, u_i, c_i]$ corresponding to i. Only a few regularization terms, e.g. the motion smoothing term (4.16) and the contact smoothing term (4.17), may depend on two or three successive frames. The problem sparsity is important to take into account, as it significantly reduces the complexity of computation from $\mathcal{O}(T^3)$ (without sparsity) to $\mathcal{O}(T)$ (using the problem sparsity).

Solving the problem. We solve the problem using the Levenberg-Marquardt algorithm. We rely on the Ceres solver [Agarwal et al., 2012], which is dedicated to solving sparse estimation problems (e.g. bundle adjustment [Triggs et al., 1999]), and on the Pinocchio software [Carpentier et al., 2019, 2015–2019] for the efficient computation of kinematic and dynamic quantities and their derivatives [Carpentier and Mansard, 2018a]. As Ceres solver only allows to define bound constraints, hence we implement our nonlinear constraints as penalties in the cost function.

Multi-stage optimization. In practice, we find that solving the optimization problem all at once usually leads to poor local minima. Instead we design a multi-stage optimization strategy taking inspiration in multi-stage optimization used for planning motion of humanoid robots [Tonneau et al., 2018b; Carpentier et al., 2017]. In detail, we solve a cascade of sub-problems composed of four stages.

In stage 1, we solve the discretized version of problem (4.1) only for the person's kinematic variables (q^{h}, $\dot{q}^{\mathrm{h}}$, $\ddot{q}^{\mathrm{h}}$) by "freezing" all variables and constraints related to the object, the ground plane, and the dynamics in Equations (4.3) and (4.4). This gives us a rough estimate of the person's 3D trajectory.

In stage 2, we recover the 3D position of the ground plane given the estimated 3D trajectory of the person and the contact states recognized from the input video sequence. In detail, we "unfreeze" the 3D position q^{g} of the ground plane and jointly solve for the trajectory of the person q^{h} and the position of the ground plane q^{g}.

Stage 3 is dedicated to initializing the object's 3D trajectory. This is achieved by solving for the object's kinematic variables (q^{o}, $\dot{q}^{\mathrm{o}}$, $\ddot{q}^{\mathrm{o}}$) under the contact constraints, while keeping the other variables fixed. Note that the location of the manipulated object varies significantly across the Handtool dataset. To address this, we sample four initialization options with different pre-defined 3D object orientations. We run stage 3 of the optimization for each initialization and pick among the four resulting solutions the one with the lowest cost.

Finally, in stage 4, we solve for the complete set of kinematic and control variables all at once, starting from the values provided by the previous stages. It is possible to continue improving the solution by pursing the aforementioned alternative descent scheme, but we found that a single pass was already sufficient to obtain good results.

Setting hyper-parameters. Hyper-parameters of our trajectory estimator, including the weights used for the cost terms, the camera model, the number of iterations, etc., are determined by following a combination of manual adjustment and a grid search: given a parameter of interest and a search grid, we run the optimization on a set of validation videos with known ground-truth 3D motion, evaluate the joint errors at every grid point, and update the hyper-parameter with the value leading to the lowest error. The same process is repeated in an iterative manner for the different hyper-parameters until the model outputs reasonable results on all the validation videos.

Run time. We report run time of trajectory optimization on a MacBook Pro 2016 (with 2.9GHz Intel Core i5 and 8GB memory). The optimization takes on average 3.23 seconds per frame. In detail, the four stages of the optimization, from stage 1 to stage 4, take on average 0.40, 0.02, 0.31 and 2.50 seconds per frame, respectively. When the pose of the object is not modeled, which is the case of one of our datasets introduced in the experimental section, the optimization is faster as stage 3 is skipped. By default, the optimization is run on the whole input video (around 100 frames in our datasets). We also provide an interface for running the optimization in a sliding window manner, which allows applying our method on longer videos.

4.4.2 Limitations

The proposed approach makes a good attempt in understanding human-object interactions from unconstrained videos. However, it still has several limitations which we discuss next.

First, our object model is currently limited to rigid, stick-like tools. Modeling other types of rigid objects, e.g. boxes, would require recognizing and modelling other object shapes. This is technically possible with our model but we leave it for future work. Recognizing and modelling interactions with non-rigid objects such as cloth is still an open challenge.

Second, the proposed method models the hand-object contact at a relatively coarse level by taking into account only a single joint location (the wrist). While this is reasonable for the type of objects considered in this thesis, it is relatively coarse for a more fine-grained manipulation tasks of smaller objects such as pencils or cups.

Third, we initialize our model with [Kanazawa et al., 2018] to provide the size and shape of the depicted person, but then use only the body skeletal rig in the

estimation stage. A mesh-based representation could be more descriptive.

Finally, our method does not consider object-object and object-ground interactions. For example, in the case of breaking concrete with a hammer, our method does not currently model the contact force exerted on the hammer by the concrete. Modeling the interactions between the object and the environment is an exciting direction of future work.

4.5 Appendix: generators of the ground contact forces

In this section, we describe the 3D-generators $g_n^{(3)}$ and the 6D-generators $g_{kn}^{(6)}$ for computing the contact forces exerted by the ground on the person joints. Remind that we model different types of contact depending on the type of the joint in contact.

We model the planar contacts between the human sole and the ground plane by fitting the point contact model (given by Eq. (4.9)) at each of the four sole vertices. For other types of ground contact, e.g. the knee-ground contact, we apply the point contact model directly at the human joint. We model the ground as a plane in Cartesian space: $G = \{p \in \mathbb{R}^3 | a^T p = b\}$, where a is a unit vector of the ground normal satisfying $a \in \mathbb{R}^3$ and $a \neq 0$, b is a scalar. We denote by μ ($\mu > 0$) the coefficient of friction between foot sole and ground.

In the following discussions, we first provide the expression of the 3D generators $g_n^{(3)}$ for modeling point contact forces and then derive the 6D generators $g_{kn}^{(6)}$ for modeling planar contact forces.

3D generators $g_n^{(3)}$ for point contact forces. Let p_k be the position of a contact point k located on the ground surface, i.e. $a^T p_k = b$. We define at contact

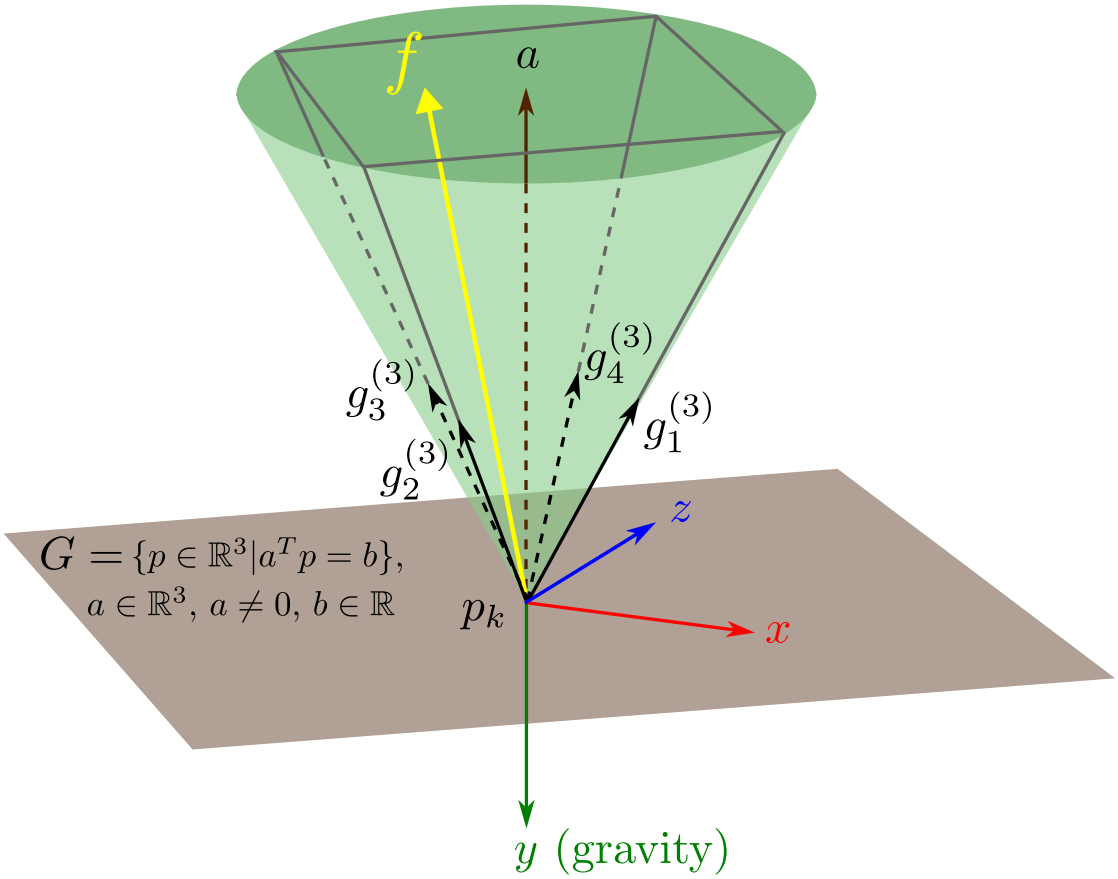


Figure 4-3: An illustration of the 3D Coulomb friction cone at the contact point p_k (the green area) and its linear approximation (the pyramid shape). The ground plane is represented by the brown area.

point k a right-hand coordinate frame C whose xz-plane overlaps the plane G and whose y-axis points towards the gravity direction, i.e., the opposite direction to the ground normal a, as shown in Figure 4-3. During point contact, it is a common assumption that the ground exerts only linear contact forces (denoted by f) at the position of contact. In other words, the spatial contact force expressed in the local frame C can be expressed as:

$$ {}^{C}\phi = \begin{pmatrix} f \\ \mathbf{0}_{3\times 1} \end{pmatrix}. \tag{4.18} $$

The 3D Coulomb friction cone and the pyramid approximation are also illustrated in Figure 4-3. To prevent the contact point from sliding, the contact force must satisfy the following *contact-stability conditions*:

- $f \cdot a > 0$, which means that the contact force f must point from the environment (ground) to the robot.

- $\|f - (f \cdot a)a\| \leq (f \cdot a)\mu$, which means that the magnitude of the tangential component of f (which includes the friction force) must not exceed the magnitude of the normal component of f, otherwise the contact switches to the sliding mode.

The second condition requires that the contact force f lies in a second-order "ice-cream" cone, or more formally a 3D Coulomb friction cone. We denote the Coulomb friction cone by the symbol $\mathcal{K}^3$.

In fact, the two contact-stability conditions can be unified into a single constraint if we define θ as half of the apex angle of $\mathcal{K}^3$ and let $\mu = \tan\theta$. The contact-stability condition is rewritten as:

$$\|f - (f \cdot a)a\| \leq (f \cdot a)\tan\theta.$$

In our case, we assume that the ground normal $a = (0, -1, 0)^T$ as shown in Figure 4-3. The contact-stability condition becomes:

$$\mathcal{K}^3 = \{f = (f_x, f_y, f_z)^T | \sqrt{f_x^2 + f_z^2} \leq -f_y \tan\theta\}.$$

It is a common practice to approximate the friction cone $\mathcal{K}^3$ by the pyramid

$$\mathcal{K}^{3\prime} = \{f = \sum_{n=1}^{4} \lambda_n g_n^{(3)} | \lambda_n \geq 0\},$$

in which the approximation is represented as the linear combination of four 3D-generators:

$$g_1^{(3)} = (\sin\theta, -\cos\theta, 0)^T, \tag{4.19}$$

$$g_2^{(3)} = (-\sin\theta, -\cos\theta, 0)^T, \tag{4.20}$$

$$g_3^{(3)} = (0, -\cos\theta, \sin\theta)^T, \tag{4.21}$$

$$g_4^{(3)} = (0, -\cos\theta, -\sin\theta)^T. \tag{4.22}$$

In other words, we are approximating the 3D Coulomb friction cone $\mathcal{K}^3$ with the conic hull $\mathcal{K}^{3\prime}$ spanned by 4 points on the boundary of $\mathcal{K}^3$, namely, $g_n^{(3)}$ with $n = 1, 2, 3, 4$. The approximation is illustrated in Figure 4-3 using the inverted pyramid with grey strokes.

6D generators $g_{kn}^{(6)}$ for planar (sole) contact forces. Here we show how to obtain the 6D generator $g_{kn}^{(6)}$ from $g_n^{(3)}$ and the contact point position p_k. As described earlier, we approximate human sole as a rectangle area with 4 contact points. We assume that the sole overlaps the ground plane G during contact. Similar to the point contact, we define 5 parallel coordinate frames C_k, one at each of the four sole contact points, plus a frame A at the ankle joint. Note that the frames C_k and A are parallel to each other, i.e., there is no rotation but only translation when passing from one frame to another. We can write the contact force at contact point k as the 6D spatial force

$${}^{C_k}\phi_k = \sum_{n=1}^{4} \lambda_{kn} \begin{pmatrix} g_n^{(3)} \\ \mathbf{0}_{3\times 1} \end{pmatrix}, \text{ with } \lambda_{kn} \geq 0. \tag{4.23}$$

We denote by ${}^{A}p_k$ the position of contact point c_k in the ankle frame A, and by ${}^{A}X_{C_k}^{*}$ the matrix converting spatial forces from frame C_k to frame A. We can then

express the contact force in frame A:

$$
\begin{aligned}
{}^{A}\phi &= \sum_{k=1}^{4} {}^{A}X^{*}_{C_k}\, {}^{C_k}\phi_k && (4.24)\\
&= \sum_{k=1}^{4} \begin{pmatrix} I_3 & {}^{A}p_k\times \\ 0_3 & I_3 \end{pmatrix}^{-T} {}^{C_k}\phi_k && (4.25)\\
&= \sum_{k=1}^{4}\sum_{n=1}^{4} \lambda_{kn} g^{(6)}_{kn}, && (4.26)
\end{aligned}
$$

where

$$
g^{(6)}_{kn} = \begin{pmatrix} g^{(3)}_n \\ {}^{A}p_k \times g^{(3)}_n \end{pmatrix}. \qquad (4.27)
$$

Chapter 5

Experiments

In this chapter, we present quantitative and qualitative evaluation of the reconstructed 3D person-object interactions. Since we focus on not only human poses but also object poses and contact forces, evaluating our results is difficult due to the lack of ground truth forces and 3D object poses in standard 3D pose benchmarks such as Ionescu et al. [2014]. Consequently, we evaluate our motion and force estimation quantitatively on a recent Biomechanics video/MoCap dataset capturing challenging dynamic parkour motions [Maldonado et al., 2017]. In addition, we report joint errors on our newly collected dataset of videos depicting handtool manipulation actions. Furthermore, we show qualitative results on both datasets to demonstrate the quality of our motion/force estimation. Finally, we discuss the main failure modes of our method.

5.1 Parkour dataset

This dataset contains RGB videos capturing human subjects performing four typical parkour actions: *kong-vault*, *moving-up*, *pull-up* and *safety-vault*. These are highly dynamic motions with rich contact interactions with the environment. Half of the

videos in the dataset are provided with ground truth 3D motion and contact forces captured with a Vicon motion capture system and force sensors. Due to the blur of fast motion in the parkour actions, this dataset is challenging for computer vision algorithms.

Evaluation set-up. We evaluate our method on the 28 parkour sequences with ground truth 3D motion and contact forces, while the remaining videos are used for training the contact recognizer. We evaluate the accuracy of the recovered 3D human poses using the common approach of computing the mean per joint position error (MPJPE) of the estimated 3D pose with respect to the ground truth after rigid alignment [Gower, 1975]. For evaluating contact forces we express the estimated and the ground truth 6D forces at the position of the contact aligned with the world coordinate frame provided in the dataset. We split the 6D force into linear and moment components and report the average Euclidean distance of the linear force and the moment with respect to the ground truth.

Results. We report joint errors for different actions in Table 5.1 and compare results with the HMR [Kanazawa et al., 2018] method, which is used to warm-start our method. To make it a fair comparison, we use the same Openpose 2D joints as input. In addition, we evaluate the SMPLify [Bogo et al., 2016] 3D pose estimation method. We also compare results with the previous version of this work [Li et al., 2019], which uses slightly different regularization of the estimated trajectory and forces. We report results for two variants of our approach. The first variant ("generic model") uses the same hyperparameters of the cost-function for all actions. The second variant ("action-specific models") uses action-specific hyperparameters adapted for each action (e.g. to regularize more strongly the motion of the legs in actions where legs are not used). Starting from the hyper-parameters of the generic model, the action-specific hyper-parameters are obtained by performing grid search,

Method	Kong-vault	Muscle-up	Pull-up	Safety-vault	Avg
SMPLify [Bogo et al., 2016]	121.75	147.41	120.48	169.36	139.69
HMR [Kanazawa et al., 2018]	111.36	140.16	132.44	149.64	135.65
Li et al. [2019]	98.42	125.21	119.92	<u>138.45</u>	122.11
Ours (generic model)	<u>93.05</u>	<u>124.55</u>	<u>101.13</u>	140.20	<u>116.13</u>
Ours (action-specific models)	**92.77**	**122.83**	**99.98**	**137.32**	**115.45**

Table 5.1: Mean per joint position error (in mm) of the recovered 3D motion for each action on the Parkour dataset.

Method	L. Sole		R. Sole		L. Hand		R. Hand	
	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)
Li et al. [2019]	144.23	23.71	138.21	22.32	107.91	131.13	113.42	134.21
Ours (generic model)	**142.11**	**22.91**	**137.34**	**20.11**	**105.07**	**130.42**	**112.21**	**132.94**

Table 5.2: Estimation errors of the contact forces exerted on soles and hands on the Parkour dataset.

as described in Section 4.4 but here using validation videos of only one action class. The results show that our generic model outperforms all the baseline methods by more than 10mm on average on this challenging data, and that our action-specific models always achieve better performance on the corresponding actions compared to the baselines.

The force estimation results are summarized in Table 5.2 where we also report results of the previous version of this work [Li et al., 2019], which produces similar results. We observe higher errors of the estimated moments at hands (compared to soles), which we believe is due to the challenging nature of the Parkour sequences where the entire person's body is often supported by hands. In this case, the hand may exert significant force and torque to support the body, and a minor shift in the force direction may lead to significant errors. In Figure 5-1, we also show an example of temporal evolution of the magnitude of the estimated linear force and torque compared with the ground truth coming from the force sensors. The estimates correspond fairly well to the ground truth. We believe the spurious peak in the estimate around frame 40 is due to the error in contact recognition, which produces a spurious linear force and a small torque.

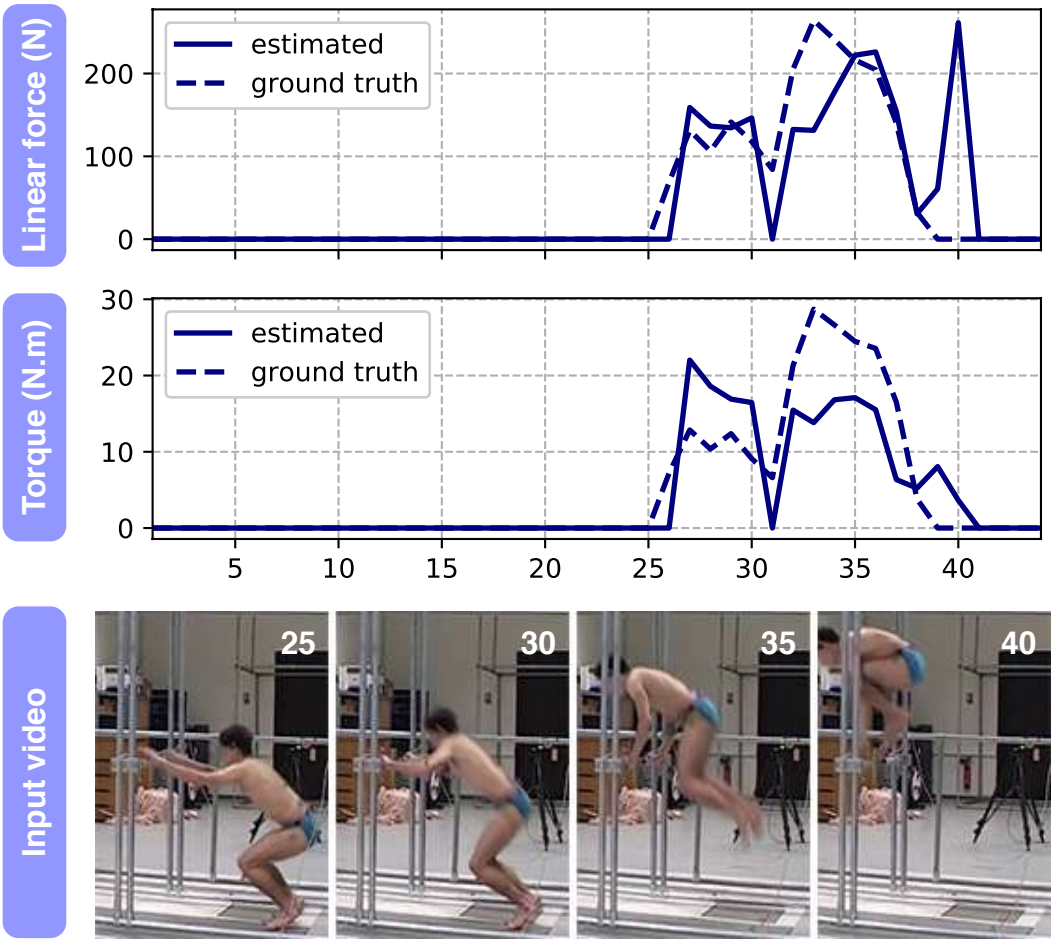


Figure 5-1: Example of temporal evolution of the magnitude of the estimated linear force (top) and torque (middle) at the person's left hand compared with the ground truth coming from the force sensors on an example sequence from the Parkour dataset. The x-axis represents time (here frame numbers). Sample frames from the sequence with their corresponding frame numbers are shown at the bottom.

5.2 Handtool dataset

In addition to the Parkour data captured in a controlled set-up, we would like to demonstrate generalization of our approach to the "in the wild" Internet instructional videos. For this purpose, we have collected a dataset of object manipulation videos, which we refer to as the Handtool dataset. The dataset contains videos of people manipulating four types of tools: *barbell*, *hammer*, *scythe*, and *spade*. For each type of tool, we chose among the top videos returned by YouTube five videos covering a range of actions. We then cropped short clips from each video showing the whole human body and the tool.

Evaluation of 3D human poses. For each video in the Handtool dataset, we have manually annotated the 3D positions of the person's left and right shoulders, elbows, wrist, hips, knees, and ankles, for the first, the middle, and the last frame. The 3D annotation is done using the Berkeley Human Annotation Tool [Bourdev and Malik, 2011], by following these three steps: (i) annotate the 2D joint locations in the image, (ii) specify the relative depth ordering for linked joints, and (iii) run the optimization approach described in Taylor [2000] to obtain a 3D stick figure. This annotation process is repeated until the 3D figure is visually correct according to the annotator. We evaluate the accuracy of the recovered 3D human poses by computing their MPJPE after rigid alignment. Quantitative evaluation of the recovered 3D poses is shown table 5.3. On average, our generic model (the same as for the Parkour dataset) outperforms all the baselines on this dataset. Our action-specific models achieve on average even better performance. Our approach achieves the best results on all individual actions except on scythe. After manual inspection of the results, we believe that this is due to the inaccuracy of the 3D model of the scythe, which is represented as a 3D line segment without explicitly modelling the handle of the scythe, which in turn affects the accuracy of the

Method	Barbell	Spade	Hammer	Scythe	Avg
SMPLify [Bogo et al., 2016]	130.69	135.03	93.43	**112.93**	118.02
HMR [Kanazawa et al., 2018]	105.04	97.18	96.34	115.42	103.49
Li et al. [2019]	104.23	95.21	95.87	114.22	102.38
Ours (generic model)	83.95	89.21	91.78	125.12	97.51
Ours (action-specific models)	**83.12**	**88.89**	**90.23**	114.13	**94.09**

Table 5.3: Mean per joint position error (in mm) of the recovered 3D human poses for each tool type on the Handtool dataset.

Method	Barbell	Hammer	Scythe	Spade
Mask R-CNN [He et al., 2017]	33/42/54	35/44/45	**63**/72/76	54/79/93
Ours (generic model)	**47/72/96**	**63/91/98**	51/**87/98**	**56/85/99**

Table 5.4: The percentage of endpoints for which the estimated 2D location lies within 25/50/100 pixels (in 600×400 pixel image) from the manually annotated ground truth location.

estimated 3D human poses (via the person-object contact model).

However, the differences between the methods are reaching the limits of the accuracy of the manually provided 3D human pose annotations on this dataset. For example, Marinoiu et al. [2013] point out that manual 3D annotation errors can range up to 100 mm per joint [Ionescu et al., 2014].

Evaluation of 2D object poses. To evaluate the quality of estimated object poses, we manually annotated 2D object endpoints in every 5th frame of each video in the Handtool dataset and calculated the 2D Euclidean distance (in pixels) between each manually annotated endpoint and its estimated 2D location provided by our method. The 2D location is obtained by projecting the estimated 3D tool position back to the image plane. We compare our results to the output of the Mask R-CNN instance segmentation baseline [He et al., 2017] (which provides initialization for our person-object interaction model). In Table 5.4 we report for

both methods the percentage of endpoints for which the estimated endpoint location lies within 25, 50, and 100 pixels from the annotated ground truth endpoint location. The results demonstrate that our approach provides in most cases more accurate and stable object endpoint locations compared to the Mask R-CNN baseline thanks to modelling the interaction between the object and the person. Lower results of our approach for scythe for the strict 25 pixel threshold can be again attributed to the inaccuracy of the 3D scythe model approximated only as a 3D line segment.

5.3 Ablation study

To gain further insight into the improvements over the conference version of this work [Li et al., 2019], we perform an ablation study of (i) the newly introduced person 3D consistency loss (4.6) (also referred to as the 3D data term) and (ii) the new force regularization term (4.17), which smooths not only the temporal variation but also the magnitude of the estimated contact forces. These experiments are done using the Parkour dataset which has precise and dense ground truth for the 3D motion and contact forces captured by MoCap and force sensors. Unless otherwise mentioned, the experiments are based on the generic model described previously.

Ablation of the 3D data term. In this ablation, we remove the 3D data term (4.6) from the generic model while keeping the rest of the cost terms and the related parameters. The results are reported in Table 5.5, where we compare the mean per joint position error (MPJPE) of the ablated model with the original generic model. The results show that on average the new 3D data term improves the 3D pose estimates, though the improvement is relatively minor. Qualitatively, we have observed that the 3D data term plays the role of a pose prior that encodes, for example, the relative depth of the different joints (e.g. between the

person's left and right hand), which is captured in the strong 3D prior of the HMR approach [Kanazawa et al., 2018].

Ablation of force regularization. The new force regularization term (4.17) smooths not only the temporal variation of the estimated contact forces [Li et al., 2019] but also the magnitude of the estimated contact forces. Therefore, we evaluate and compare two ablated models against our generic model. In the first ablated model (Ours (no force regularization)), we remove both terms regularizing the temporal variation and the magnitude of the estimated contact forces (i.e. the second and the third term in Eq. (4.17)). In the second ablated model (Ours (no $\|f_k\|^2$ in (4.17))), we only remove the third term regularizing the magnitude of the estimated contact forces, i.e. this model regularizes only the temporal variation of the estimated contact forces. Note that this form of force regularization was used in the conference version of this work [Li et al., 2019]. Quantitative results are reported in Table 5.6 and clearly show the benefit of regularizing both the temporal variation and the magnitude of the estimated contact forces (Ours (generic model)), which results in the lowest errors. While we cannot compute force estimation errors on the Handtool dataset due to the lack of ground truth data, we can still perform a simple ablation analysis by plotting the temporal variation of the estimated linear forces and torques with and without the force regularization terms. This is shown on an example video sequence for the left-hand contact force in Figure 5-2. Please note how the regularization of both the temporal variation and magnitude of the estimated forces (4.17) effectively smoothes the estimated forces reducing their abrupt temporal changes and unrealistic magnitudes. Figure 5-2(d) also compares the output of our model with and without force regularization at two example frames. In particular, frame #10 corresponds to the case where the model without force regularization outputs a linear force with an unrealistic

orientation and magnitude (highlighted with a bold yellow line in the image) whereas the regularized model outputs a more realistic force estimate in terms of both the orientation and magnitude. Similarly, for frame #51 the model with force regularization outputs a torque with a smaller and hence more realistic magnitude.

Method	Kong-vault	Muscle-up	Pull-up	Safety-vault	Avg
Ours (no 3D data term)	94.69	**124.12**	103.87	141.88	117.55
Ours (generic model)	**93.05**	124.55	**101.13**	**140.20**	**116.13**

Table 5.5: Ablation of the 3D data term (4.6). We report the mean per joint position error (MPJPE) in mm of the estimated 3D human motion for each action on the Parkour dataset. The first row corresponds to the ablated model, where the person 3D data term has been removed from the generic model. The second row corresponds to the generic model.

Method	L. Sole		R. Sole		L. Hand		R. Hand	
	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)	lin. force (N)	moment (N·m)
Ours (no force regularization)	148.74	86.54	144.12	79.45	128.55	137.45	133.79	136.43
Ours (no $\|f_k\|^2$ in (4.17))	143.76	23.78	139.60	22.19	109.10	133.29	117.89	133.87
Ours (generic model)	**142.11**	**22.91**	**137.34**	**20.11**	**105.07**	**130.42**	**112.21**	**132.94**

Table 5.6: Ablation of force regularization terms (eq. (4.17)). We report estimation errors of contact forces exerted on soles and hands in the Parkour dataset. The first row corresponds to the ablated model where both terms regularizing the temporal variation of the force and the force magnitude are removed from our generic model. The second row corresponds to the ablated model where the term regularizing the magnitude of the estimated force is removed. The third row corresponds to our generic model.

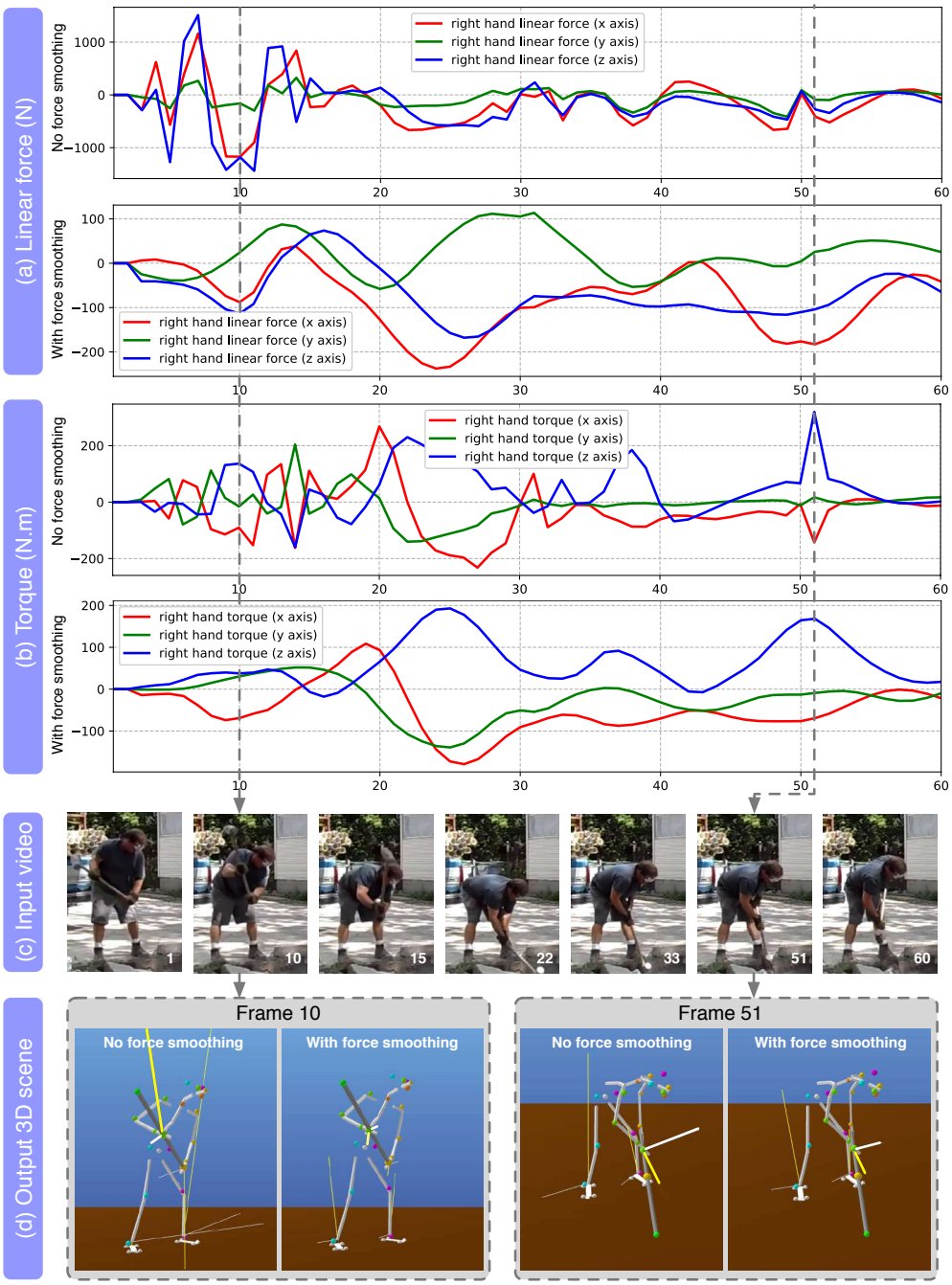


Figure 5-2: Plots of the estimated linear contact force **(a)** and torque **(b)** at the right hand for an example video from the Handtool dataset. In all plots the x-axis represents time (in frame numbers). In both (a) and (b), the top plot is without force regularization and the bottom plot is with force regularization (i.e. the generic model). **(c)** shows example frames with their corresponding frame numbers. **(d)** shows the estimated 3D scene at two sample frames with the highlighted linear contact force (bold yellow) and torque (bold white) at the right hand. Please note how force regularization effectively smoothes the estimated forces and torques reducing their unrealistic abrupt temporal changes and large magnitudes (note the different scales of the y-axis in the different plots).

5.4 Qualitative results

Here we show qualitative examples. Additional video results are available on our Project webpage [2021].

Figure 5-3 shows a collection of qualitative results at sampled video frames in the Handtool (top four rows) and the Parkour (bottom four rows) datasets. For each sample, we first show the original frame (left image), followed by the estimated 3D motion and forces from the original viewpoint (middle image), and the same 3D scene from a different viewpoint (right image). Note that for the Parkour dataset we recognize the contact states of human joints but do not recognize and model the pose of the object (the metal construction) the person is interacting with. In addition to results for individual frames from different videos, we provide in Figure 5-4 results for two sequences of frames to demonstrate the continuity of the reconstructed actions. The sequences demonstrate that the outputs of our method are temporally consistent and smooth.

Figure 5-5 shows a comparison of our model with the baseline HMR approach [Kanazawa et al., 2018]. In the first example (hammering action), the person's hands holding the hammer are restricted to be on the handle by our contact model, thus reducing the depth ambiguity compared to 3D human poses provided by the baseline HMR [Kanazawa et al., 2018] estimator, which often outputs open arms. The second example shows an "outlier" frame of a Parkour video where HMR fails to estimate correct human body orientation due to heavy occlusion and motion blur. In this case, our method relies on the model of dynamics and the pose prior to synthesize the person's motion in between good predictions. Due to these reasons, we observe that our method often predicts better poses than the baseline methods that are applied to individual frames and do not model the temporal interaction between the person and the tool.

The qualitative results also demonstrate that our model predicts reasonable

contact forces. The directions of the contact forces exerted on the person's hands are consistent with the object's motion trajectory and gravity, and the ground reaction forces generally point towards the direction opposite to gravity. Specifically, in the video with the person practicing back squat with barbell (see the left example in the second row of Figure 5-3), the reconstructed object contact forces and ground reaction forces are distributed evenly on the person's hands, and knees, respectively. Another example is scythe (third row of Figure 5-3, right), where the distribution of ground reaction forces at the person's feet follows the swings of the body while cutting the grass. In the shown frame the person's center of mass is above their right leg, leading to larger contact force at the right leg.

5.5 Failure modes

During the experiments we found three typical cases in which the estimation stage of the proposed approach may fail to output good results. The failure modes are illustrated in Figure 5-6.

Missing object 2D endpoint detection. The estimated object 2D endpoints are often very noisy due to heavy occlusions between human limbs and the manipulated object. To solve this problem, we filter out the detected endpoints with low confidence scores at the output of the recognition stage. However, this measurement may produce missing observations in the estimated 2D endpoint sequences. An extreme example is shown in Figure 5-6 in the first row, where there is no predicted endpoint at all. This is because the barbell handle is completely occluded as seen in the input frame on the left. In this case, our contact motion model can infer the position of the barbell handle from the position of hands, but the results are often not very accurate.

Contact recognition errors. The second row of Figure 5-6 shows an example with incorrectly estimated contact state. In this case, the person's right knee is incorrectly recognized as not in contact. This has lead to incorrect force estimation in the output.

Incorrectly localized human joints in the image. We observe that our method struggles to estimate correct 3D poses if the quality of 2D detection is too low. An example is shown in the bottom row of Figure 5-6 where the 2D detection of the person's left foot is missing. This has lead to errors in estimating the 3D location of the person's left leg.

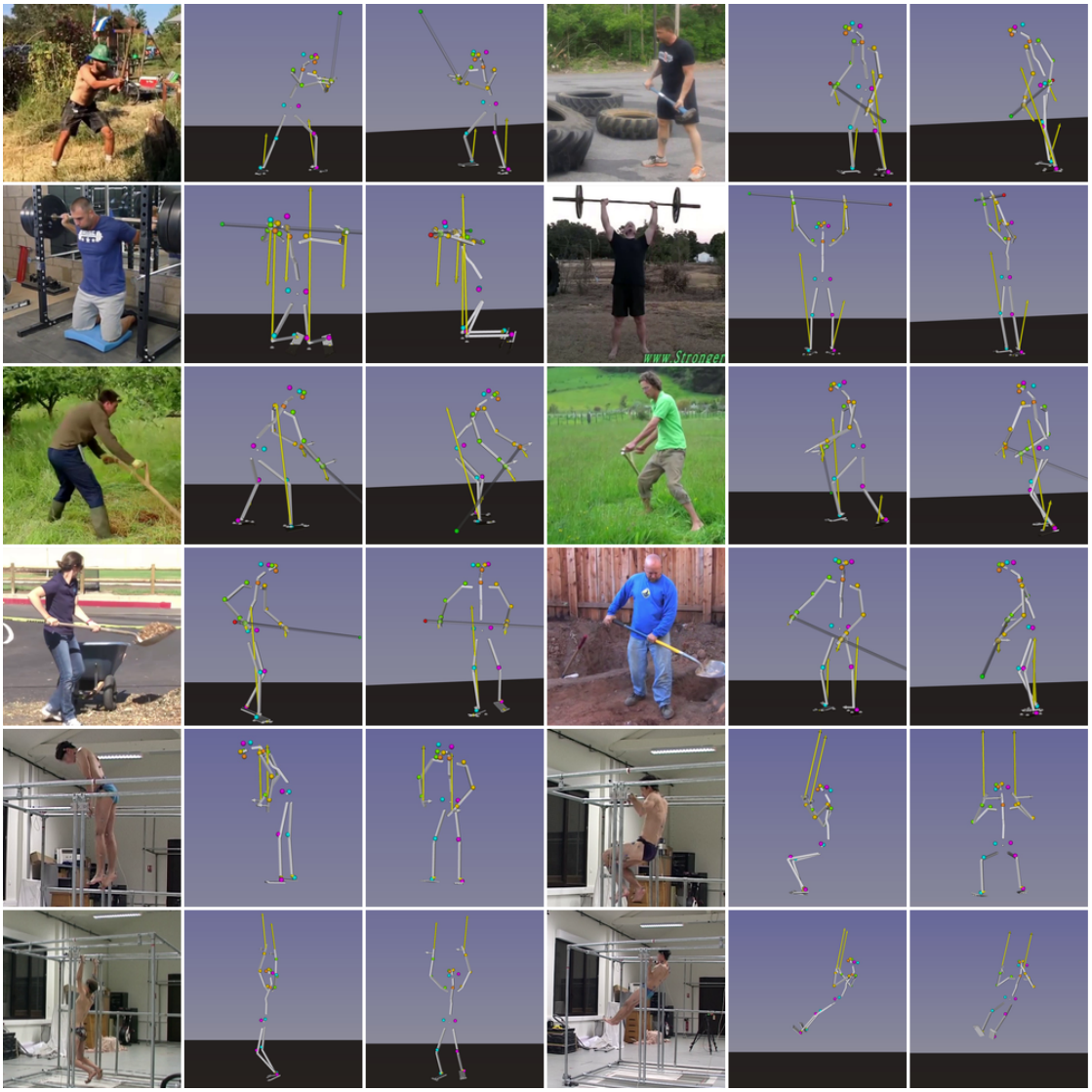


Figure 5-3: Example qualitative results on the Handtool (rows 1-4) and Parkour (rows 5-6) datasets. **Rows from top to bottom**: hammer, barbell, scythe, spade, muscle-up and pull-up. Each example shows the input frame (left) and two different views of the output 3D pose of the person and the object (middle, right). The yellow and the white arrows in the output show the contact forces and moments, respectively. The length of the arrow represents the magnitude of the force normalized by gravity.

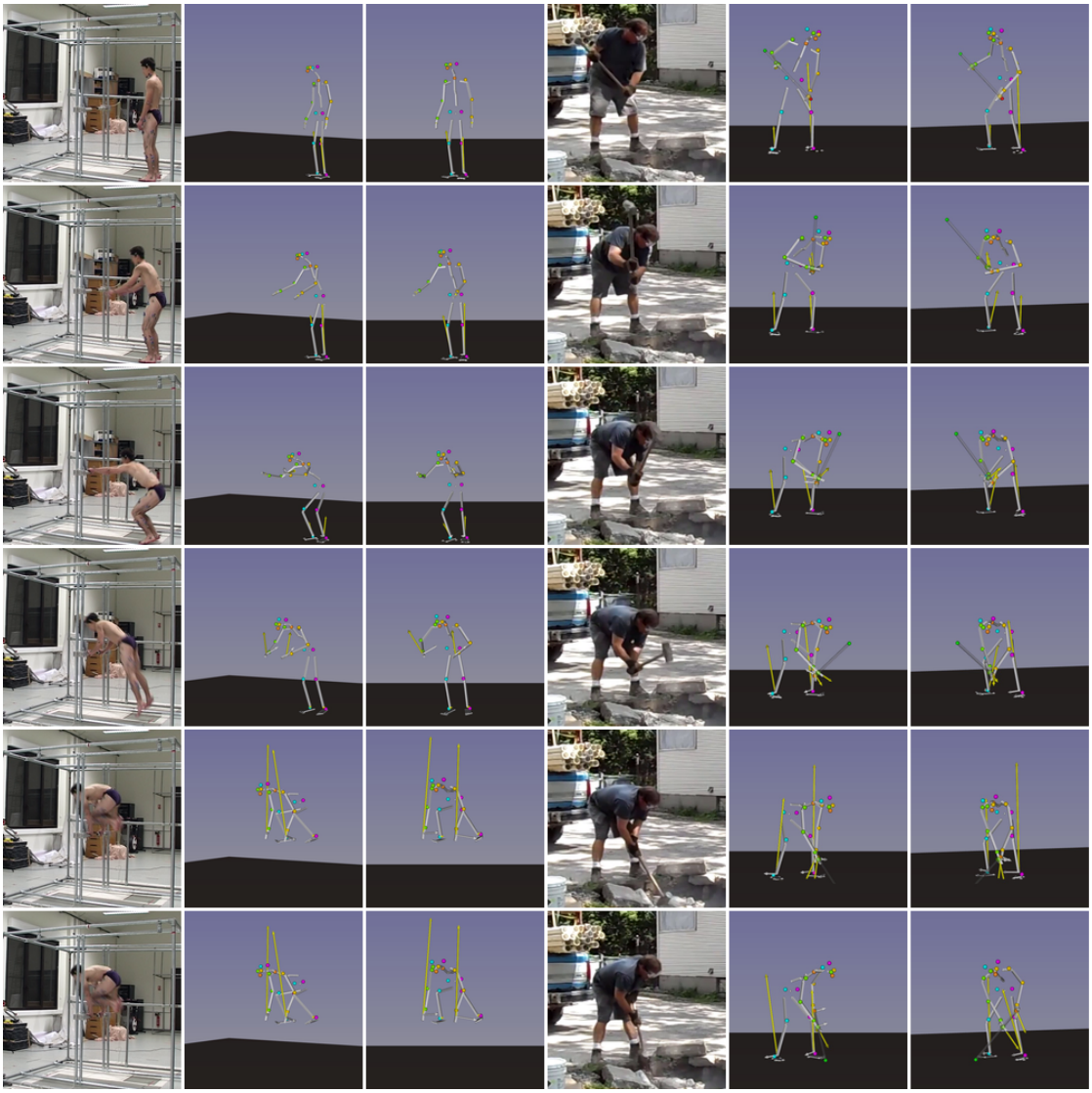
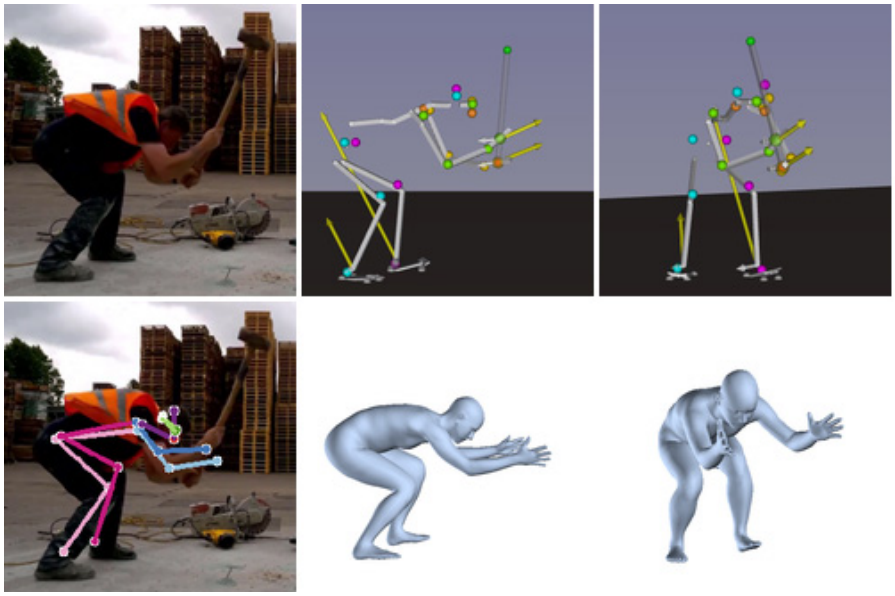
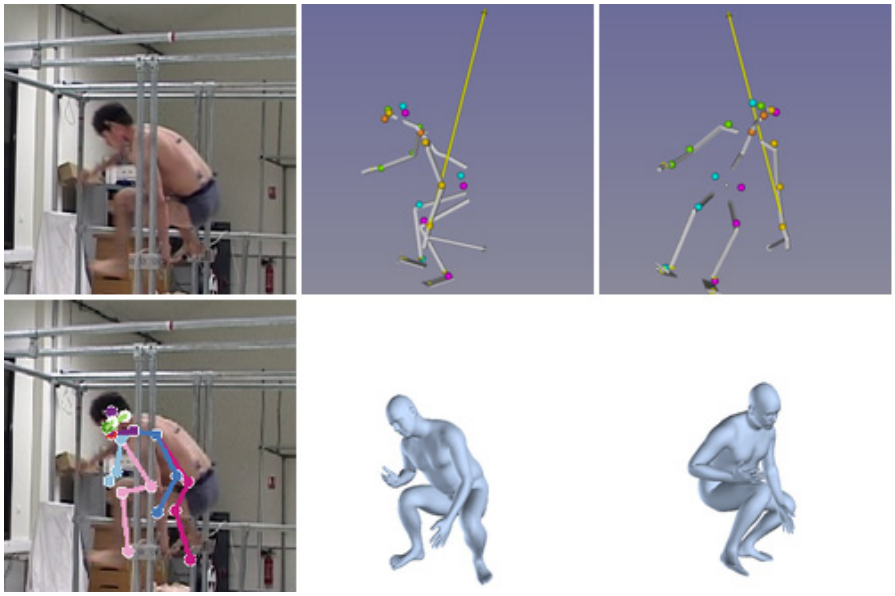


Figure 5-4: Example qualitative results on image sequences. **Columns 1-3**: muscle-up (Parkour dataset), **Columns 4-6**: hammer (Handtool dataset). Additional video results are available at the Project webpage [2021].



(a) Hammering example



(b) Parkour example

Figure 5-5: Qualitative comparison with the baseline HMR estimator [Kanazawa et al., 2018]. In each example, the top row shows the input frame (left) and the output of our method from two different viewpoints (middle, right). The bottom row shows the estimated 2D joints (left) and the output of the HMR baseline shown from two different viewpoints (middle, right). In the hammering example (a) the person's hands holding the hammer are restricted to be on the handle by our contact model, thus reducing the depth ambiguity compared to 3D human poses provided by the baseline HMR [Kanazawa et al., 2018] estimator, which often outputs open arms. The example (b) shows an "outlier" frame of a Parkour video where HMR fails to estimate correct human body orientation w.r.t the camera due to heavy occlusion and motion blur.

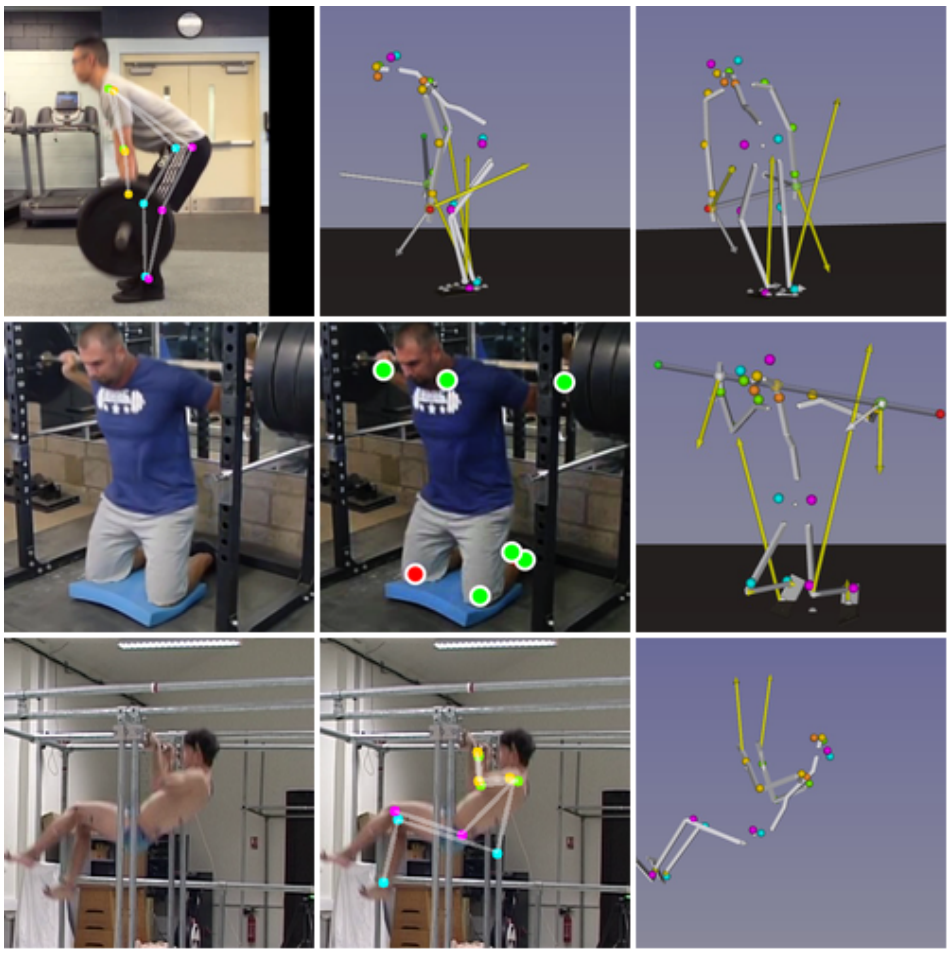


Figure 5-6: Main failure modes of our method. **Top row**: missing object 2D endpoint detection. The handle of the barbell is not detected, which affects the 3D output of our model. **Middle row**: contact recognition errors. The person's right knee is incorrectly recognized as not in contact (red), leading to incorrect force estimates shown on the right. **Bottom row**: incorrect 2D human joints. The missing 2D detection of the person's left foot has lead to errors in estimating the 3D location of the left leg.

Chapter 6

Discussion and future work

In this final chapter, we summarize the main contributions of the thesis and discuss possible research directions for future work.

6.1 Contributions of the thesis

In this thesis, we have developed a visual recognition system which automatically reconstructs the 3D dynamic scene depicting a human subject interacting with a tool given a single RGB video. In particular, the system takes as input video frames together with a simple object model, and outputs a 3D motion of the person and the object including contact forces and torques actuated by the human limbs. The major contributions are summarized below:

- In Chapter 3, we have developed models and methods for extracting three types of 2D measurements from the input RGB video: (i) the 2D positions of a set of predefined human joints, (ii) the 2D endpoint positions of a stick-like object, and (iii) the contact states of the human joints that can potentially touch the object. These models are collectively referred to as the recognition stage of the proposed system.

- In Chapter 4, we have developed a model for reliably estimating the 3D person-object motion and contact forces from the 2D measurements obtained in the recognition stage. In particular, we rely on numerical optimization to jointly estimate the person-object 3D motion and forces under physics constraints. The research problem is formulated as a large-scale trajectory optimization problem to model the dynamics of 3D movement of the person, the manipulated object, and their interaction with created and broken physical contacts.

- In Chapter 5, we have validated our approach on a recent video MoCap dataset with ground truth contact forces. In addition, we have collected a new dataset of unconstrained instructional videos depicting people manipulating different objects and have demonstrated benefits of our approach on this data. Our work opens up the possibility of large-scale learning of human-object interactions from Internet instructional videos.

6.2 Future work

In this final part, we analyze possible future research directions for improving the work presented in this thesis. We divide the discussion into two parts based on the two stages of our proposed system.

Improving the 2D recognition stage.

- **Estimation of human 2D poses.** We have redesigned the strategy of the bottom-up parsing step in the original implementation of Openpose to deal with the interference of human instances in background. The method has been shown to be successful overall on the testing data, but may fail to track the subject of interest due to occlusion, e.g., of half of a body. We propose to

investigate more recent human pose estimators, for example, PARE [Kocabas et al., 2021], which has been shown to be robust to heavy occlusion.

- **Estimation of object 2D keypoints.** The proposed object 2D keypoint estimation module requires the objects to be of a stick-like shape, which is a relatively strong assumption. One possible way to overcome this limitation is to investigate more recent object estimators that are able to determine more than two keypoints, e.g. CosyPose [Labbe et al., 2020] and MegaPose [Labbé et al., 2022]. This interesting direction has already been investigated by some recent work [Zorina et al., 2021].

- **Contact recognition.** Another interesting direction is improving the quality of contact recognition. This can be done by scaling up the training data to fine-tune the contact recognition networks. Alternatively, we can leverage more recent contact recognizers, such as Shan et al. [2020], which models hand-object contact and trains a hand contact recognizer on a much larger dataset. In Cao et al. [2021], the authors propose a method to recognize more different types of hand-object contact from 2D images.

Improving the 3D person-object trajectory estimation.

- **More generic 3D object model.** Our object model is currently limited to rigid, stick-like tools. This assumption is relatively strong, which makes our approach hard to be applied to actions involving objects with big volume, such as moving a cardboard box. Modeling other types of rigid objects, e.g. boxes, would require recognizing and modelling other object shapes. This direction is very interesting and technically possible with our model, but we leave it for future work.

  It is also very interesting to model non-rigid objects such as cloth. However,

recognizing and modelling interactions with non-rigid objects is still an open challenge.

- **Human body representation.** In the estimation stage of our approach, we use the body skeletal rig to represent the size and shape of the depicted person. As a possible future direction, using a mesh-based representation instead would be more descriptive than the skeletal rig considered in this thesis.

- **Modeling different types of contact.** In the proposed approach, we model the hand-object contact at a relatively coarse level by taking into account only a single joint location (the wrist). Although this is reasonable for the type of stick-like objects considered in this thesis, it is too coarse for a more fine-grained manipulation of smaller objects such as pencils or cups. The next step would be handling more sophisticated contact motion such as grasping a paper or a box, etc.

  Finally, our method does not consider object-object and object-ground contacts. For example, in the case of breaking concrete with a hammer, our method does not currently model the contact force exerted on the hammer by the concrete. Modeling the contacts and interactions between the object and the environment is an exciting research topic that could be further investigated for future work.

Bibliography

Abdulla, W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN, 2017. 26

Agarwal, S., Mierle, K., and Others. Ceres solver. http://ceres-solver.org, 2012. 53

Akhter, I. and Black, M. J. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 2015. 16

Alayrac, J.-B., Bojanowski, P., Agrawal, N., Laptev, I., Sivic, J., and Lacoste-Julien, S. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 20

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 24

Biegler, L. T. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, volume 10, chapter 10. Siam, 2010. 52

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016. 16, 18, 46, 62, 63, 67

Boulic, R., Thalmann, N. M., and Thalmann, D. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6(6):344–358, Nov 1990. ISSN 1432-2315. doi: 10.1007/BF01901021. URL https://doi.org/10.1007/BF01901021. 20

Bourdev, L. and Malik, J. The human annotation tool. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/shape/hat/, 2011. 66

Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *CVPR*, 2016. 19

Brubaker, M. A., Fleet, D. J., and Hertzmann, A. Physics-based person tracking using simplified lower-body dynamics. In *CVPR*, 2007. 18

Brubaker, M. A., Sigal, L., and Fleet, D. J. Estimating contact dynamics. In *CVPR*, 2009. 18

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 17, 23, 33

Cao, Z., Radosavovic, I., Kanazawa, A., and Malik, J. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12417–12426, 2021. 83

Carpentier, J. and Mansard, N. Analytical derivatives of rigid body dynamics algorithms. In *Robotics: Science and Systems*, 2018a. 53

Carpentier, J. and Mansard, N. Multi-contact locomotion of legged robots. *IEEE Transactions on Robotics*, 2018b. 47, 48

Carpentier, J., Valenza, F., Mansard, N., et al. Pinocchio: fast forward and inverse dynamics for poly-articulated systems. https://stack-of-tasks.github.io/pinocchio, 2015–2019. 53

Carpentier, J., Del Prete, A., Tonneau, S., Flayols, T., Forget, F., Mifsud, A., Giraud, K., Atchuthan, D., Fernbach, P., Budhiraja, R., et al. Multi-contact locomotion of legged robots in complex environments–the loco3d project. In *RSS Workshop on Challenges in Dynamic Legged Locomotion*, page 3p, 2017. 53

Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiraux, F., Stasse, O., and Mansard, N. The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, 2019. 53

Chen, C.-H. and Ramanan, D. 3d human pose estimation= 2d pose estimation+ matching. In *CVPR*, 2017. 16

Delaitre, V., Sivic, J., and Laptev, I. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011. 18

Diehl, M., Bock, H., Diedam, H., and Wieber, P.-B. Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In *Fast Motions in Biomechanics and Robotics*. Springer, 2006. 20

Doumanoglou, A., Kouskouridas, R., Malassiotis, S., and Kim, T.-K. 6d object detection and next-best-view prediction in the crowd. In *CVPR*, 2016. 19

Featherstone, R. *Rigid body dynamics algorithms*. Springer, 2008. 51

Fouhey, D. F., Delaitre, V., Gupta, A., Efros, A. A., Laptev, I., and Sivic, J. People watching: Human actions as a cue for single view geometry. *IJCV*, 110 (3):259–274, 2014. 18

Gall, J., Rosenhahn, B., Brox, T., and Seidel, H.-P. Optimization and filtering for human motion capture. *IJCV*, 87(1-2):75, 2010. 16

Gammeter, S., Ess, A., Jäggli, T., Schindler, K., Leibe, B., and Van Gool, L. Articulated multi-body tracking under egomotion. In *ECCV*, 2008. 16

Gower, J. C. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 62

Grabner, A., Roth, P. M., and Lepetit, V. 3D Pose Estimation and 3D Model Retrieval for Objects in the Wild. In *CVPR*, 2018. 19

Gupta, A., Kembhavi, A., and Davis, L. S. Observing human-object interactions: Using spatial and functional compatibility for recognition. *PAMI*, 31(10):1775–1789, 2009. 18

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016. 29

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870. 26, 67

Herdt, A., Perrin, N., and Wieber, P.-B. Walking without thinking about it. In *International Conference on Intelligent Robots and Systems (IROS)*, 2010. doi: 10.1109/IROS.2010.5654429. 20

Hinterstoisser, S., Lepetit, V., Rajkumar, N., and Konolige, K. Going further with point pair features. In *ECCV*, 2016. 19

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 17

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, jul 2014. 17, 61, 67

Jiang, Y., Koppula, H., and Saxena, A. Hallucinated humans as the hidden context for labeling 3d scenes. In *CVPR*, 2013. 18

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 16, 18, 46, 55, 62, 63, 67, 69, 73, 78

Kanazawa, A., Zhang, J. Y., Felsen, P., and Malik, J. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019. 16

Kocabas, M., Athanasiou, N., and Black, M. J. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 18

Kocabas, M., Huang, C.-H. P., Hilliges, O., and Black, M. J. PARE: Part attention regressor for 3D human body estimation. In *Proc. International Conference on Computer Vision (ICCV)*, pages 11127–11137, Oct 2021. 83

Kuffner, J., Nishiwaki, K., Kagami, S., Inaba, M., and Inoue, H. Motion planning for humanoid robots. In *Robotics Research. The Eleventh International Symposium*, 2005. 20

Labbe, Y., Carpentier, J., Aubry, M., and Sivic, J. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 83

Labbé, Y., Manuelli, L., Mousavian, A., Tyree, S., Birchfield, S., Tremblay, J., Carpentier, J., Aubry, M., Fox, D., and Sivic, J. Megapose: 6d pose estimation of novel objects via render & compare. In *CoRL 2022-Conference on Robot Learning*, 2022. 83

Li, Y., Wang, G., Ji, X., Xiang, Y., and Fox, D. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *ECCV*, 2018. 19

Li, Z., Sedlar, J., Carpentier, J., Laptev, I., Mansard, N., and Sivic, J. Estimating 3d motion and forces of person-object interactions from monocular video. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 12, 62, 63, 64, 67, 68, 69

Li, Z., Sedlar, J., Carpentier, J., Laptev, I., Mansard, N., and Sivic, J. Estimating 3d motion and forces of human–object interactions from internet videos. *International Journal of Computer Vision*, 130(2):363–383, 2022. 12

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312. 26

Loing, V., Marlet, R., and Aubry, M. Virtual training for a real application: Accurate object-robot relative localization without calibration. *IJCV*, Jun 2018. ISSN 1573-1405. doi: 10.1007/s11263-018-1102-6. URL https://doi.org/10.1007/s11263-018-1102-6. 26

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34 (6):248, 2015. 39, 46

Loper, M. M., Mahmood, N., and Black, M. J. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 33(6):220:1–220:13, Nov 2014. URL http://doi.acm.org/10.1145/2661229.2661273. 46

Maldonado, G. Some biomechanical and robotic models. https://github.com/GaloMALDONADO/Models, 2018. 42

Maldonado, G., Bailly, F., Souères, P., and Watier, B. Angular momentum regulation strategies for highly dynamic landing in Parkour. *Computer Methods in Biomechanics and Biomedical Engineering*, 20(sup1):123–124, 2017. doi: 10.1080/10255842.2017.1382892. URL https://hal.archives-ouvertes.fr/hal-01636353. 61

Malmaud, J., Huang, J., Rathod, V., Johnston, N., Rabinovich, A., and Murphy, K. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015. 20

Marinoiu, E., Papava, D., and Sminchisescu, C. Pictorial human spaces: How well do humans perceive a 3d articulated pose? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1289–1296, 2013. 67

Martinez, J., Hossain, R., Romero, J., and Little, J. J. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 16

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 20

Miech, A., Alayrac, J.-B., Smaira, L., Laptev, I., Sivic, J., and Zisserman, A. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 20

Mordatch, I., Todorov, E., and Popović, Z. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4): 43, 2012. 20, 21

Moreno-Noguer, F. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017. 16

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 17

Newell, A., Huang, Z., and Deng, J. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, 2017. 17

Oberweger, M., Rad, M., and Lepetit, V. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *ECCV*, 2018. 19

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017. 16

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143:1–143:14, Jul 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201311. URL http://doi.acm.org/10.1145/3197517.3201311. 21, 22

Posa, M., Cantu, C., and Tedrake, R. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014. 20

Prest, A., Ferrari, V., and Schmid, C. Explicit modeling of human-object interactions in realistic videos. *PAMI*, 35(4):835–848, 2013. 18

Project webpage. https://www.di.ens.fr/willow/research/motionforcesfromvideo/, 2021. 73, 77

Rad, M. and Lepetit, V. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017. 19

Rad, M., Oberweger, M., and Lepetit, V. Feature Mapping for Learning Fast and Accurate 3D Pose Inference from Synthetic Images. In *CVPR*, 2018. 19

Rempe, D., Guibas, L. J., Hertzmann, A., Russell, B., Villegas, R., and Yang, J. Contact and human dynamics from monocular video. In *European Conference on Computer Vision*, pages 71–87. Springer, 2020. 18

Schultz, G. and Mombaur, K. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on mechatronics*, 15(5):783–792, 2010. 20

Shan, D., Geng, J., Shu, M., and Fouhey, D. Understanding human hands in contact at internet scale. In *Proc. CVPR*, 2020. 83

Shimada, S., Golyanik, V., Xu, W., and Theobalt, C. Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020. 18

Shimada, S., Golyanik, V., Xu, W., Pérez, P., and Theobalt, C. Neural monocular 3d human motion capture with physical awareness. *ACM Transactions on Graphics (ToG) - SIGGRAPH*, 2021. 18

Sidenbladh, H., Black, M. J., and Fleet, D. J. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000. 16

Tassa, Y., Erez, T., and Todorov, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012. doi: 10.1109/IROS.2012.6386025. 20

Taylor, C. J. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80(3): 349–363, 2000. 66

Tejani, A., Tang, D., Kouskouridas, R., and Kim, T.-K. Latent-class hough forests for 3d object detection and pose estimation. In *ECCV*, 2014. 19

Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016. 16

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. *CoRR*, abs/1703.06907, 2017. URL http://arxiv.org/abs/1703.06907. 26

Tonneau, S., Del Prete, A., Pettré, J., Park, C., Manocha, D., and Mansard, N. An Efficient Acyclic Contact Planner for Multiped Robots. *IEEE Transactions on Robotics (TRO)*, 2018a. doi: 10.1109/TRO.2018.2819658. 20

Tonneau, S., Del Prete, A., Pettré, J., Park, C., Manocha, D., and Mansard, N. An efficient acyclic contact planner for multiped robots. *IEEE Transactions on Robotics*, 34(3):586–601, 2018b. 53

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, 1999. 53

Wei, X. and Chai, J. Videomocap: Modeling physically realistic human motion from monocular video sequences. *ACM Trans. Graph.*, 29(4):42:1–42:10, Jul 2010. ISSN 0730-0301. doi: 10.1145/1778765.1778779. URL http://doi.acm.org/10.1145/1778765.1778779. 18, 19

Westervelt, E. R., Grizzle, J. W., and Koditschek, D. E. Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control*, 48(1):42–56, 2003. doi: 10.1109/TAC.2002.806653. 20

Winkler, A. W., Bellicoso, C. D., Hutter, M., and Buchli, J. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018. 20

Xiang, D., Joo, H., and Sheikh, Y. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10974, 2019. 16, 17

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017. URL http://arxiv.org/abs/1711.00199. 19

Yao, B. and Fei-Fei, L. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *PAMI*, 34(9): 1691–1703, 2012. 18

Zanfir, A., Marinoiu, E., and Sminchisescu, C. 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*, pages 2148–2157, 2018. 18

Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. Sparseness meets deepness: 3d human pose estimation from monocular video. In *CVPR*, 2016. 16, 18

Zorina, K., Carpentier, J., Sivic, J., and Petrík, V. Learning to manipulate tools by aligning simulation to video demonstration. *IEEE Robotics and Automation Letters*, 7(1):438–445, 2021. 7, 83

RÉSUMÉ

Dans cette thèse, nous étudions le problème de la reconstruction automatique en 3D des mouvements d'une personne agissant dans une scène complexe avec un objet, à partir d'une seule vidéo RVB. Nous développons une méthode complète pour établir une correspondance entre les images vidéo 2D et une interprétation 3D de la scène, qui est représentée par les poses 3D de la personne et de l'objet manipulé, les positions des contacts avec l'objet et avec l'environnement, et les forces de contact exercées à ces interfaces. Les principales contributions de cette thèse sont les suivantes. Dans un premier temps, nous introduisons une approche pour estimer conjointement le mouvement et les forces impliqués dans la vidéo en formulant un problème d'optimisation avec contrainte de trajectoire minimisant une fonction de perte, composite, intégrée dans le temps. Deuxièmement, nous développons une méthode pour reconnaître automatiquement à partir de la vidéo d'entrée la position 2D et les instants de contact entre la personne et l'objet ou le sol. Troisièmement, nous validons expérimentalement notre approche sur un jeu de données vidéo-MoCap récent capturant des actions typiques de parkour et équipé de forces et de trajectoires de vérité au sol.

MOTS CLÉS

Vision par ordinateur, robotique, apprentissage profond, estimation de pose 3D.

ABSTRACT

In this thesis, we investigate the problem of automatically reconstructing the 3D dynamic scene depicting a person interacting with a tool in a single RGB video. The objective is to obtain a 3D interpretation of the scene represented by the 3D poses of the person and the manipulated object over time, the contact positions and the contact forces exerted on the human body. The main contributions of this thesis are as follows. First, we introduce an approach to jointly estimate the motion and the actuation forces of the person on the manipulated object by modeling the contacts and the dynamics of the interactions. Second, we develop a method to automatically recognize from the input video the 2D position and timing of contacts between the person and the object or the ground. Third, we validate our approach on a recent video-MoCap dataset capturing typical parkour actions and equipped with ground truth forces and trajectories.

KEYWORDS

Computer vision, robotics, deep learning, 3D pose estimation.