Learning human body and human action representations from visual data

Gül Varol

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Learning human body and human action representations from visual data

Soutenue par

Gül VAROL

Le 29 Mai 2019

École doctorale n°386

Sciences Mathématiques de Paris Centre

Spécialité

Informatique

Composition du jury :

Francis BACH Inria	*Président du jury*
Iasonas KOKKINOS University College London	*Rapporteur*
Marc POLLEFEYS ETH Zurich	*Rapporteur*
Andrew ZISSERMAN University of Oxford	*Examinateur*
Ivan LAPTEV Inria	*Directeur de thèse*
Cordelia SCHMID Inria	*Directrice de thèse*

ENS | PSL

Abstract

The focus of visual content is often people. Automatic analysis of people from visual data is therefore of great importance for numerous applications in content search, autonomous driving, surveillance, health care, and entertainment.

The goal of this thesis is to learn visual representations for human understanding. Particular emphasis is given to two closely related areas of computer vision: human body analysis and human action recognition.

In human body analysis, we first introduce a new synthetic dataset for people, the SURREAL dataset, for training convolutional neural networks (CNNs) with free labels. We show the generalization capabilities of such models on real images for the tasks of body part segmentation and human depth estimation. Our work demonstrates that models trained only on synthetic data obtain sufficient generalization on real images while also providing good initialization for further training. Next, we use this data to learn the 3D body shape from images. We propose the BodyNet architecture that benefits from the volumetric representation, the multi-view re-projection loss, and the multi-task training of relevant tasks such as 2D/3D pose estimation and part segmentation. Our experiments demonstrate the advantages from each of these components. We further observe that the volumetric representation is flexible enough to capture 3D clothing deformations, unlike the more frequently used parametric representation.

In human action recognition, we explore two different aspects of action representations. The first one is the discriminative aspect which we improve by using long-term temporal convolutions. We present an extensive study on the spatial and temporal resolutions of an input video. Our results suggest that the 3D CNNs should operate on long input videos to obtain state-of-the-art performance. We further extend 3D CNNs for optical flow input and highlight the importance of the optical flow quality. The second aspect that we study is the view-independence of the learned video representations. We enforce an additional similarity loss that maximizes the similarity between two temporally synchronous videos which capture the same action. When used in conjunction with the action classification loss in 3D CNNs, this similarity constraint helps improving the generalization to unseen viewpoints.

In summary, our contributions are the following: (i) we generate photo-realistic synthetic data for people that allows training CNNs for human body analysis, (ii) we propose a multi-task architecture to recover a volumetric body shape from a single image, (iii) we study the benefits of long-term temporal convolutions for human action recognition using 3D CNNs, (iv) we incorporate similarity training in multi-view videos to design view-independent representations for action recognition.

Résumé

Le contenu visuel se concentre souvent sur les humains. L'analyse automatique des humains à partir de données visuelles revêt donc une grande importance pour de nombreuses applications dans la recherche de contenu, la conduite autonome, la surveillance, la santé et le divertissement.

Le but de cette thèse est d'apprendre des représentations visuelles pour l'analyse des humains. Un accent particulier est mis sur deux domaines étroitement liés de la vision artificielle : l'analyse du corps humain et la reconnaissance des actions humaines.

Dans l'analyse du corps humain, nous introduisons tout d'abord un nouvel ensemble de données synthétiques sur les personnes, l'ensemble de données SURREAL, destiné à l'entraînement de réseaux de neurones convolutionnels (CNN) sans avoir recours à des annotations manuelles. Nous montrons les capacités de généralisation de tels modèles sur des images réelles pour les tâches de segmentation des parties du corps et d'estimation de leurs profondeurs. Notre travail démontre que les modèles entraînés uniquement sur des données synthétiques obtiennent une généralisation suffisante sur des images réelles tout en offrant une bonne initialisation. Ensuite, nous utilisons ces données pour apprendre la forme 3D du corps à partir d'images. Nous proposons l'architecture BodyNet qui tire parti de la représentation volumétrique, de la fonction de coût de re-projection multi-vue et d'une formulation multitâche incluant l'estimation de pose 2D / 3D et la segmentation des parties du corps. Nos expériences démontrent les avantages de chacune de ces composantes. Nous observons en outre que la représentation volumétrique est assez flexible pour capturer les déformations 3D des vêtements, contrairement à la représentation paramétrique plus fréquemment utilisée.

Dans la reconnaissance de l'action humaine, nous explorons deux aspects différents des représentations d'action. Le premier est l'aspect discriminant que nous améliorons en utilisant des convolutions temporelles à long terme. Nous présentons une étude approfondie sur les résolutions spatiales et temporelles de la vidéo d'entrée. Nos résultats suggèrent qu'utiliser des CNN 3D sur de longues vidéos d'entrée permet d'obtenir des performances de pointe. Nous étendons ensuite les CNNs 3D, nous ajoutons le flux optique en entrée et soulignons l'importance de sa qualité. Le deuxième aspect que nous étudions est l'indépendance des représentations vidéo apprises par rapport au changement de point de vue. Nous ajoutons une fonction de coût de similarité supplémentaire qui maximise la similarité entre deux vidéos temporellement synchrones qui capturent la même action. Lorsqu'elle est utilisée conjointement avec la perte de classification d'action dans les CNN 3D, cette contrainte de similarité contribue à améliorer la généralisation aux nouveaux points de vue.

En résumé, nos contributions sont les suivantes : (i) nous générons des données synthétiques photoréalistes de personnes permettant l'entraînement de CNNs pour l'analyse du corps humain, (ii) nous proposons une architecture multitâche permet-

tant d'obtenir une représentation volumétrique du corps à partir d'une seule image, (iii) nous étudions les avantages des convolutions temporelles à long terme pour la reconnaissance de l'action humaine à l'aide de CNNs 3D, (iv) nous incorporons une fonction de coût de similarité des vidéos multi-vues pour concevoir des représentations invariantes au changement de vue pour la reconnaissance d'action.

Acknowledgments

Ivan and Cordelia, I would like to thank you for your greatest support during my PhD. I feel extremely lucky to have you both as my advisors. I am especially indebted for your academic advice and guidance. Thanks Ivan for making me excited about new research ideas, for your perfectionism and constant enthusiasm. Thanks Cordelia for closely guiding me to make the right decisions, and for teaching me to ask the right questions. Working with both of you has been a great pleasure. Every meeting has been motivating, stimulating, and fun. Thank you for your invaluable time, attention, dedication, and inspiration.

I am grateful to the ERC grant ACTIVIA for generously funding my research. I would like to thank Inria Paris for providing an excellent research environment.

Many thanks go to Andrew Zisserman, Francis Bach, Iasonas Kokkinos, and Marc Pollefeys for agreeing to take part in my thesis jury.

I would like to thank additionally to Andrew Zisserman for welcoming me at Oxford. I am looking forward to working with you and learning from you.

I have been lucky to have collaborated with Michael Black. Thanks Michael for hosting me many times in your wonderful lab in Tübingen. You are a great leader and a great role model.

Thanks to Duygu Ceylan for being the kindest mentor. I learned a lot from you. I am looking forward to having more collaborations with you.

I owe a sincere thank you to Albert Ali Salah for encouraging me to start a PhD in the first place.

All members of WILLOW and SIERRA, thanks for the exceptional lab environment. Thanks to Francis Bach and Jean Ponce for leading these teams. Thanks to Josef Sivic, for giving me the opportunity to have an enjoyable teaching experience, it has been a pleasure to work with you. Anton, Christoph, Igor, Nastya, Yana, thanks for sharing offices and deadline moments with me. Thanks to the lab members for all the fun memories: Adrien Taylor, Alessandro Rudi, Alexandre De-

fossez, Anastasia Podosinnikova, Andrei Bursuc, Antoine Miech, Antoine Recanati, Anton Osokin, Aymeric Dieuleveut, Bumsub Ham, Christophe Dupuy, Damien Garreau, Damien Scieur, Dmitrii Ostrovskii, Dmitry Babichev, Dmitry Zhukov, Fabian Pedregosa, Gauthier Gidel, Guilhem Chéron, Guillaume Seguin, Ignacio Rocco, Igor Kalevatykh, Jean-Baptiste Alayrac, Julia Peyre, Justin Carpentier, Lénaic Chizat, Loïc Esteve, Loucas Pillaud Vivien, Makarand Tapaswi, Margaux Bregere, Mathieu Aubry, Matthew Trager, Maxime Oquab, Mihai Dusmanu, Minsu Cho, Nicolas Flammarion, Pascal Germain, Pierre Gaillard, Piotr Bojanowski, Rafael Rezende, Raphael Berthier, Relja Arandjelović, Rémi Leblond, Robert Gower, Robin Strudel, Ronan Riochet, Sergey Zagoruyko, Sesh Kumar, Sofiane Allayen, Suha Kwak, Tatiana Shpakova, Théophile Dalens, Thomas Eboli, Thomas Kerdreux, Tuan-Hung Vu, Vadim Kantorov, Van Huy Vo, Vijay Kumar, Vincent Roulet, Yana Hasson, Yann Labbé, and Zongmian Li.

I would like to thank my friends, some of whom are far away, but close thanks to technology! Asya, Başak, Ece, İlayda, İlker, and Ömür, thanks for all the visits to Paris.

Last but not least, I am deeply grateful to my family for their love and support. I wish to give special thanks to Umut for always being with me.

Contents

Chapter 1

Introduction

This thesis addresses human understanding from visual data with a focus on human body analysis (Part I) and human action recognition (Part II). This chapter presents the goals, the motivations, and the contributions of our work.

1.1 Goals

Our main goal is to automatically analyze human body and human motion from images. In particular, we are interested in the low-level task of estimating the 3D human body shape and the high-level task of action recognition.

The first focus of this thesis is to extract human body structure from an input image. We study body pose estimation, body part segmentation, body depth estimation, and more importantly *3D body shape estimation* problems. Figure 1-1(a) shows example input-output pairs for the body estimation task. In particular, we address the data scarcity problem by artificially generating images for training our models. We employ convolutional neural networks (CNNs) as our learning machines. While being powerful models, CNNs require large amounts of labeled data for successful learning. As we will discuss in Section 1.3, acquiring such data is challenging for our tasks. Therefore, we investigate the use of large-scale *synthetic data* for training

Figure 1-1 – (a) Sample images from the UP dataset [Lassner et al., 2017] together with their semi-automatic ground truth 3D body shapes. (b) Sample video frames from HMDB51 [Kuehne et al., 2011], UCF101 [Soomro et al., 2012], and NTU RGB+D [Shahroudy et al., 2016] action recognition datasets. HMDB51 covers 51 actions mostly collected from movies. UCF101 has 101 classes collected from YouTube. NTU RGB+D records 60 actions in lab settings.

CNNs.

Second, we develop visual representations that capture *long-term* dynamics in the video data while having discriminative property to classify human actions. In simplest words, *action recognition* refers to categorizing an input video into one or more of the predefined set of action classes, such as jumping, running etc. Here, the definition of the actions are task-dependent which we will discuss as part of the challenges (Section 1.3). Figure 1-1(b) illustrates sample actions from the datasets we work with. Our objective is to investigate the learning of long-term video representations by considering space-time convolutional neural networks. Besides the discriminative property of these video representations, we are also interested in *view-independence*. With other words, we would like similar actions to generate similar video representations independently of camera viewpoints.

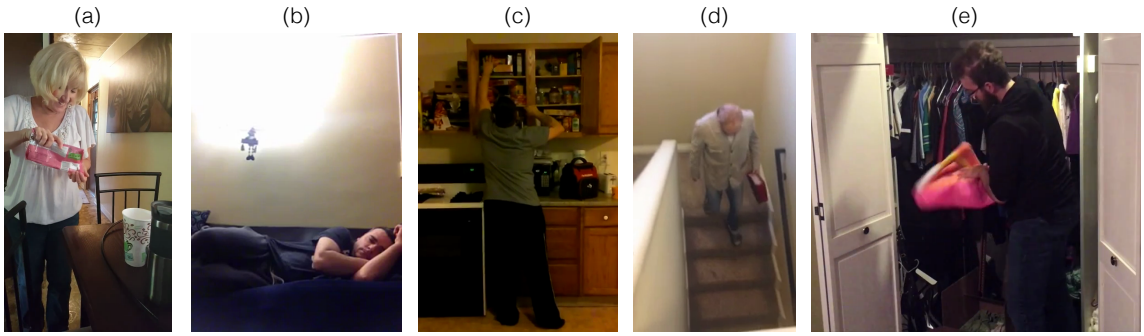


Figure 1-2 – People are the central actors of most multimedia content. We show sample video frames from the Charades dataset [Sigurdsson et al., 2016b] (see Annex A) illustrating actions in home contexts. Some actions involve interactions with objects (a) or environments (d). Temporal information is needed to distinguish some scenes, e.g. putting or taking (c).

1.2 Motivations

Computer vision research has become increasingly crucial given the rapid growth of visual data. Cisco predicts that the Internet video traffic (business and consumer, combined) will be 80% of all Internet traffic by 2022, up from 70% in 2017. It also predicts that 3 trillion minutes (5 million years) of video content will cross the Internet each month by 2022. That is 1.1 million minutes of video streamed or downloaded every second[1].

A fair amount of research in computer vision focuses on person-related tasks. These tasks include person detection, person tracking, human pose estimation, human action recognition, and others. Moreover, certain body parts receive special attention such as faces and hands. In the following, we list various reasons why human understanding from visual data is an active research area together with specific application areas.

[1] https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html

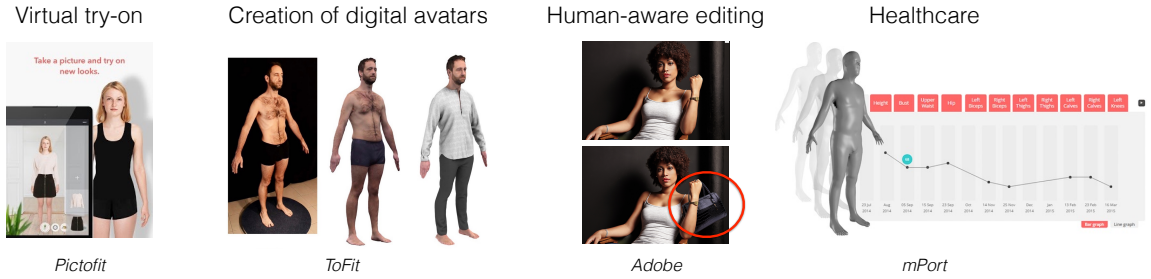


Figure 1-3 – Human body analysis has applications in various domains including fashion industry, entertainment, editing, and healthcare. Visuals are borrowed from the entities below each image.

Why human understanding? Understanding the content of visual data is the core of computer vision. The content can often be divided into several components: scene, object(s), and actor(s). Consider the Figure 1-2(c) which shows a *person* who is *putting* an *object* in the *kitchen* cupboard. One needs to understand all these components to describe the whole scene. *People* are, arguably, the central actors of most multimedia content. Therefore, understanding people in images and videos contributes to the broader goal of understanding visual data. Recent analysis [Laptev, 2013] showed that about 35% of pixels in popular consumer videos (YouTube videos, movies, TV programs) contained people. This strong bias towards people suggests that *person*[2] is a specific object category that requires more focus from computer vision researchers.

Applications of body analysis. Certain applications require the recovery of 3D body shapes from images. Body shape recovery can either be the goal on its own or can serve as a building block for other tasks. In human-computer interaction, it is often desirable that human body parts, especially hands, are detected and tracked with cheap sensors such as RGB cameras. Automatic measurement of bodies has applications in medicine and graphics. For healthcare, body shape reconstruction using

[2]In this thesis, we use *person* interchangeably with *human* although their definitions are not the same. Moreover, in this context, person is sometimes referred to as *object* although the definition of object in modern philosophy might disagree.

camera(s) can facilitate the fast recording of accurate body measurements. On the other hand, virtual try-on (see Figure 1-3) can improve the online shopping experience for clothes. For editing applications such as the Adobe Photoshop software, it is important to provide the users with human-aware image and video editing tools (see Figure 1-3 where a bag is inserted between the arm and the shoulder of the person). This requires automatic labeling of body parts as well as the depth ordering between them. Next, we review applications that benefit from a higher level understanding of visual content, i.e., understanding human motion and semantic actions.

Applications of action recognition. Analyzing people and their actions in videos has a wide range of applications. The current growth of data raises the need for effective indexing methods to enable multimedia search. In the context of security, one may need to search video recordings of surveillance cameras that span months of data which are impractical to watch. On the other hand, Internet users search video-sharing websites such as YouTube, which contains huge amounts of data. On a smaller scale, people generate increasingly more multimedia using their ubiquitous cameras. Automatic organization of holiday photos or summarization of long video archives are becoming necessary. The fast processing capacity of machines remove the need for humans to watch entire sequences for analyzing large video databases.

Another aspect of video understanding is for real-time applications. Autonomous cars should accurately localize and recognize the pedestrians and other people in their surroundings to be safely used in cities. Real-time recognition of person movements such as fall detection enables assisted living applications. Action recognition is also of high interest within entertainment industry with PlayStation Camera being a concrete example that is used to track and recognize human movements to play games. Besides entertainment, such devices are used for enriching the human-computer interaction experience, which often requires also extracting precise human body configuration in 3D.

This section presented potential applications of human body estimation and human action recognition from visual data. Next, we discuss some of the challenges which justify why these topics raise open research questions.

1.3 Challenges

The human visual system has no difficulty identifying human bodies and understanding their movements. Yet, this is currently a difficult task for machines. Cameras capture a set of pixels which are impossible to be interpreted by computers without the models that translate this raw data into meaningful representations. In this thesis, we focus on *learning* representations using convolutional neural networks. The artificial learning of human understanding is an active research field due to the following challenges.

People are deformable objects. Unlike a rigid chair or a rigid TV screen, the human body is highly articulated. The pose of a rigid object can be represented with 6-dimensional (6 DOF) parameters; however, the pose of a human body is higher dimensional. The limbs, i.e. arms and legs, move with many degrees of freedom. These movements cause *self-occlusions* when captured with a monocular camera. That causes some parts of the body not being observed in the data. Moreover, the body deforms depending on the configuration of the body parts, i.e. the *pose*. Due to these properties, human bodies are difficult to model; therefore, difficult to represent. There is also a high variability in body shapes which makes it challenging to gather data covering all these variations.

Representation problem. Several approximations are used when modeling human bodies. The most common representation is the skeleton representation consisting of a sparse set of joints. These joints are predefined points corresponding to certain locations on the body, such as elbows, wrists, knees. A typical skeleton def-

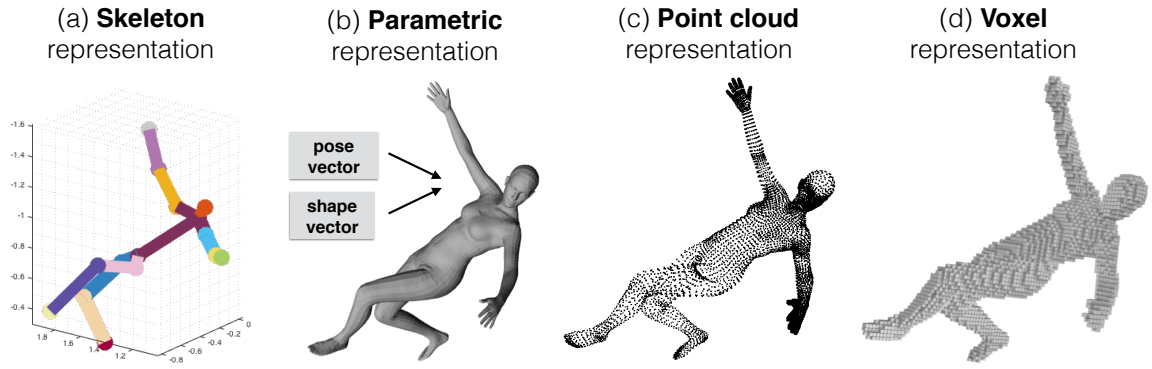


Figure 1-4 – How to optimally represent human bodies is an open research problem. We illustrate several alternatives: (a) the common skeleton representation that encodes the joint locations, (b) the parametric representation that, given a low-dimensional parameterization, generates the body mesh, e.g. SMPL model [Loper et al., 2015], (c) the point cloud representation that encodes the vertex locations, (d) the voxel representation over a discrete occupancy grid.

inition is depicted in Figure 1-4(a). Skeleton representation is simple and sufficient for some applications. However, for many applications, knowledge of the 3D human body shape, which is far more expressive than a set of sparse points, is required. In this case, how to represent dense 3D shapes for learning frameworks such as CNNs remains an open problem. Figure 1-4(b-d) demonstrates several alternatives, each of which has various advantages and disadvantages. We will discuss more on this topic in Chapter 4.

There is not enough annotated data. Equipped with large amounts of labeled training data, CNNs obtain excellent performance on some computer vision tasks, such as object recognition. Manual labeling at large scale for certain tasks, however, is prohibitively expensive. Here, we discuss two aspects of the data scarcity problem in the context of human analysis.

The first challenge concerns acquiring 3D ground truth for human bodies. Manually annotating images with precise 3D is a difficult task even for humans. Millimetric accuracy is possible with the help of additional sensors such as depth cameras and

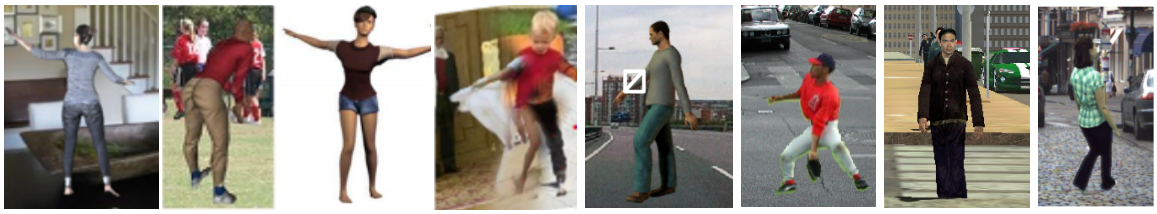


Figure 1-5 – Sample synthetic images taken from [Varol et al., 2017; Chen et al., 2016b; Ghezelghieh et al., 2016; Rogez and Schmid, 2016; Okada and Soatto, 2008; Pishchulin et al., 2012; Marin et al., 2010; Pishchulin et al., 2011] from left to right.

motion capture markers. However, these sensors limit the variability of the captured data since they are not prevalent. Datasets with ground truth are typically gathered in a motion capture studio with the same background and a few subjects. It is well known that learning from limited data has problems of generalization. A recently popular alternative has been the use of synthetic data which can be generated using graphics tools for rendering diverse datasets. However, a recipe for how to create a good synthetic dataset for training is still unknown. Figure 1-5 shows sample synthetic images from various datasets. Chapter 3 details our approach for addressing this issue.

The second aspect relates to the complexity of the human motion recognition problem. Compared to the object recognition community which has witnessed significant improvements with the release of the large-scale ImageNet dataset [Deng et al., 2009], action recognition has received moderate gains with CNNs. Despite the active effort of gathering large-scale video datasets (Sports-1M [Karpathy et al., 2014], ActivityNet [Caba Heilbron et al., 2015], Charades [Sigurdsson et al., 2016b], Kinetics [Kay et al., 2017], AVA [Gu et al., 2018], VLOG [Fouhey et al., 2018]) the optimal pipeline for collecting training datasets for action recognition remains challenging. First, the action recognition problem is more complex due to the additional temporal dimension, requiring even bigger datasets. Second, annotating temporal data is time-consuming, therefore not scalable. The aforementioned aspects make the human body and motion analysis particularly challenging.

Figure 1-6 – The appearance of an action differs due to many factors, such as pose, clothing, subject, background clutter, occlusion, motion blur, and lighting. We illustrate this effect by varying these factors for the same *jumping* action.

Variability. A natural challenge in computer vision is the variation caused by some factors such as imaging conditions. The algorithms should be robust against the clutter, occlusions, and lighting variations. The same scene appears drastically different when captured from different viewpoints. Enforcing view-independence in our learned representations is difficult, especially when the available training data has certain bias towards certain viewpoints. For example, people tend to film themselves from the frontal view, while computer vision algorithms need to work from the sides. We will present more details on view-independence in Chapter 6. Variations introduced specifically by human appearances also include different types of clothing, different body poses and shapes. Regarding the actions, different actors typically perform the same action with various speeds and styles. This kind of intra-class variation makes the action classification particularly challenging. Figure 1-6 presents several of these variations with examples.

(a) running / playing football (b) cutting tomatoes / cooking (c) *riding* a bike (d) *riding* a horse

Figure 1-7 – How to define actions for recognition is a challenge. Samples (a-b) illustrate the granularity problem. Samples (c-d) illustrate the difficulty of associating natural language to actions.

How do we define actions? An intrinsic problem with action recognition research is the definition of actions. First, which level of granularity is required? For Figure 1-7(a), the person can be running or playing football. For Figure 1-7(b), the person can be cooking or cutting tomatoes. Second, is it possible to attribute text (e.g. one verb, one phrase) to one action? If we consider verbs as actions, *riding* a horse becomes the same action as *riding* a bike although the associated movements differ, as in Figure 1-7(c-d). Although some attempts have been made to address these issues by defining a taxonomy of classes or by defining a composite vocabulary of actions [Nagel, 1988; Bobick, 1997; Hongeng and Nevatia, 2001; Gonzàlez, 2004; Fernández et al., 2011], how to model the hierarchical and the ambiguous nature of actions remains challenging. We refer the reader to [Shapovalova et al., 2011] for a review on different taxonomies.

1.4 Contributions

In the following, we list the publications contributions, as well as the software and dataset releases that were performed during the course of this thesis. We will detail the contributions within four of the publications in Chapters 3-6.

1.4.1 Publications

The work done during this PhD led to the following publications:

- G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *CVPR*, 2017. [Varol et al., 2017] (Chapter 3)
- G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018. [Varol et al., 2018a] (Chapter 4)
- G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *TPAMI*, 2018. [Varol et al., 2018b] (Chapter 5)
- G. Varol, I. Laptev, and C. Schmid. On view-independent video representations for action recognition. *Work in progress*, 2019. [Varol et al., 2019] (Chapter 6)
- G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. [Sigurdsson et al., 2016b] (Annex A)
- Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. [Hasson et al., 2019] (Annex B)

1.4.2 Software and dataset contributions

Software. The code for three chapters of this thesis are publicly released:

- BodyNet: The code and pre-trained models for body shape estimation are released as part of the project presented in [Varol et al., 2018a] (Chapter 4). https://github.com/gulvarol/bodynet

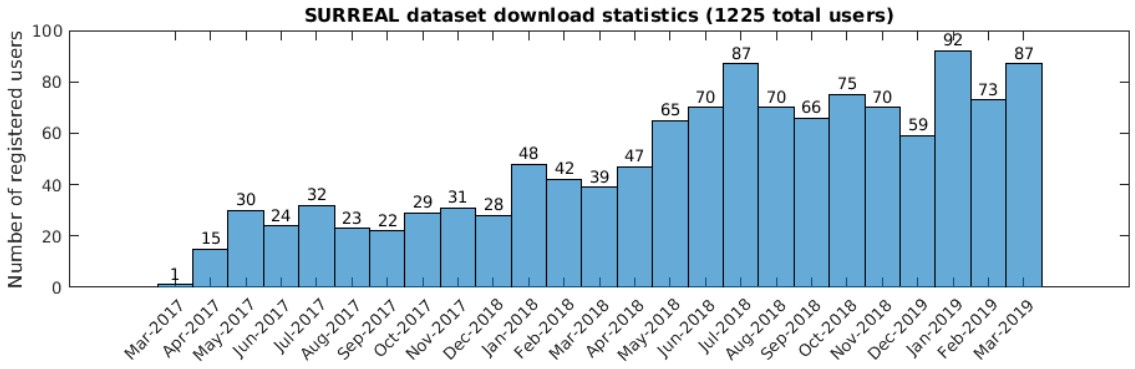


Figure 1-8 – The download statistics of our SURREAL dataset since the public release in March 2017. More than 1000 users registered to download the dataset in two years.

- SURREAL: The code to generate synthetic person videos, the documentation for the SURREAL dataset, as well as the code and pre-trained models for body segmentation and depth estimation are released as part of the project presented in [Varol et al., 2017] (Chapter 3). `https://github.com/gulvarol/surreal`

- LTC: The code and pre-trained models for action recognition with 3D convolutional neural networks are released as part of the project presented in [Varol et al., 2018b] (Chapter 5). `https://github.com/gulvarol/ltc`

SURREAL dataset. We have publicly released the SURREAL dataset (`https://www.di.ens.fr/willow/research/surreal/data/`) with the publication of [Varol et al., 2017] (Chapter 3) in collaboration with Max Planck Institute. The name stands for 'Synthetic hUmans foR REAL tasks'. SURREAL is the first large-scale person dataset to generate depth, body parts, optical flow, 2D/3D pose, body model parameters, body mesh for RGB video input. The dataset contains 6M frames of synthetic humans. The images are photo-realistic renderings of people under large variations in shape, texture, viewpoint and pose. To ensure realism, the synthetic bodies are created using the SMPL body model [Loper et al., 2015], whose parameters are fit by the MoSh [Loper et al., 2014] method given raw 3D MoCap marker data. The dataset has been downloaded more than 1000 times to date (see Figure 1-8) and

is used for many other research publications on human body analysis.

ObMan dataset. We have publicly released the ObMan dataset (https://www.di.ens.fr/willow/research/obman/) with the publication of [Hasson et al., 2019] (Annex B) in collaboration with Max Planck Institute. ObMan consists of synthetic images of hands manipulating objects. The generative hand model MANO [Romero et al., 2017] is used to simulate object grasps and to render realistic training data.

Charades dataset. We have publicly released the Charades dataset (https://allenai.org/plato/charades/) with the publication of [Sigurdsson et al., 2016b] (Annex A) in collaboration with Carnegie Mellon University and Allen Institute for Artificial Intelligence. Charades is a large-scale dataset with a focus on common household activities collected using the 'Hollywood in Homes' approach. The name comes from a popular American word guessing game where one player acts out a phrase and the other players guess what phrase it is. In a similar spirit, we asked hundreds of people from Amazon Mechanical Turk to act out a paragraph that we presented to them. The workers additionally provide action classification, localization, and video description annotations. The dataset contains 9,848 videos of daily activities 30.1 seconds long on average. The videos contain interactions with 46 object classes and have a vocabulary of 30 verbs leading to 157 action classes. It has 66,500 temporally localized actions, 12.8 seconds long on average, recorded by 267 people in three continents. We believe this dataset can provide a crucial stepping stone in developing action representations, learning object states, human object interactions, modeling context, object detection in videos, video captioning and many more.

1.5 Outline

This thesis consists of seven chapters including this introduction. The main content is divided into two parts. We also include an annex.

Literature survey. Chapter 2 reviews the related works in the literature with particular focus on (i) human body analysis, and (ii) human action recognition.

Part I: Human body analysis. We present two contributions in Chapters 3 and 4 on body analysis from images.

In Chapter 3, we focus on the use of synthetic data for training CNNs. We generate a large-scale video dataset of people (SURREAL) and train neural networks for the tasks of human body part segmentation and human depth estimation. We show the generalization capabilities of models trained only on synthetic data on real images. We study three scenarios: training (i) only on real data, (ii) only on synthetic data, and (iii) first on synthetic data, then on real data. We conclude that the third scenario is effective, suggesting that synthetic data provide good initialization for CNNs. We also show that the scenario (ii) performs surprisingly well, suggesting that the SURREAL dataset reflects real distributions. In case of low real data regime, (ii) even outperforms (i). We further provide some analysis on the ingredients of synthetic data generation, such as diverse clothing and diverse motion, to gain insights about how to create training images of people.

In Chapter 4, we use this data to learn the 3D human body shape estimation task. The emphasis of this chapter is on the effectiveness of the volumetric representation for learning body shapes with CNNs. We present our approach, called BodyNet, which predicts a 3D occupancy grid given a single image of a person. Experiments on our network architecture demonstrate the advantages of guiding 3D shape estimation with intermediate tasks such as 2D/3D pose and 2D segmentation in a multi-task framework. The BodyNet approach also benefits from an additional re-projection loss that constrains the output shape to spatially align with the 2D segmentation. This in return helps obtaining more confident predictions of near-surface voxels, as well as voxels belonging to the extremities such as hands and feet. We observe that the volumetric representation is flexible and has the potential to capture 3D clothing

deformations, unlike the more frequently used alternative of parametric representation.

Part II: Video representations for human action recognition. We present two contributions in Chapters 5 and 6 on action recognition from videos.

In Chapter 5, we learn video representations using 3D convolutional neural networks for human action recognition. A video input is a 3D spatio-temporal object which makes 3D convolution a natural operation for such input. However, we show that 3D CNNs outperform 2D CNNs only when their input spans long-term video. Here, long-term refers to ~100 frames (which is still short, i.e., 4 seconds), as opposed to the frequently used alternative of ~10 frames. This chapter presents an experimental study on the importance of the temporal dimension of video. Moreover, we investigate the use of optical flow as input in 3D CNNs. We observe that the quality of action recognition is proportional to the quality of the the optical flow algorithm. By combining long-term RGB video and long-term optical flow video inputs to 3D CNNs, we obtain state-of-the-art performance on the standard action recognition datasets UCF101 and HMDB51.

In Chapter 6, we focus on the view-independence aspect of learned video representations. We enforce an additional loss on the 3D CNN training for action recognition. This loss maximizes the similarity between two temporally synchronous videos which capture the same action. We observe that enforcing such constraints between viewpoints available in the training data improves generalization to unseen viewpoints in the test data. This chapter also introduces a scenario where the training data only has one viewpoint with action labels. In this case, we perform cross-dataset experiments in which an auxiliary dataset is provided with multiple viewpoints but without action labels. Our experiments demonstrate the significant gains achieved with our similarity loss defined only on the auxiliary dataset.

Discussion. We conclude in Chapter 7 with a summary of contributions, a discussion of open problems and future work.

Annex. In the annex, we include our work on crowdsourcing data collection for action recognition (Annex A) and our work on joint hand-object reconstruction (Annex B).

Chapter 2

Literature Review

This chapter will survey work on human body analysis (Section 2.1) and human action recognition (Section 2.2).

2.1 Human body analysis

In this section, we will review previous research on computer vision tasks about extracting human body related information, given a single RGB image. We start by articulated pose estimation (Section 2.1.1) which usually corresponds to estimating the pixel locations of body joints on the image (2D pose estimation) or the 3D joint locations typically in camera coordinates (3D pose estimation). Next, we briefly review the tasks which require denser outputs such as body part segmentation (Section 2.1.2), body depth estimation (Section 2.1.3), and body shape estimation (Section 2.1.4). Segmentation assigns a body segment such as head, torso etc. for each image pixel. Similarly, depth estimation assigns, for each image pixel, a metric value for the distance from the camera. Finally, we discuss the recent literature on the body shape estimation task that seeks to produce a dense representation for the 3D body surface. Figure 2-1 illustrates each of these tasks on a sample image.

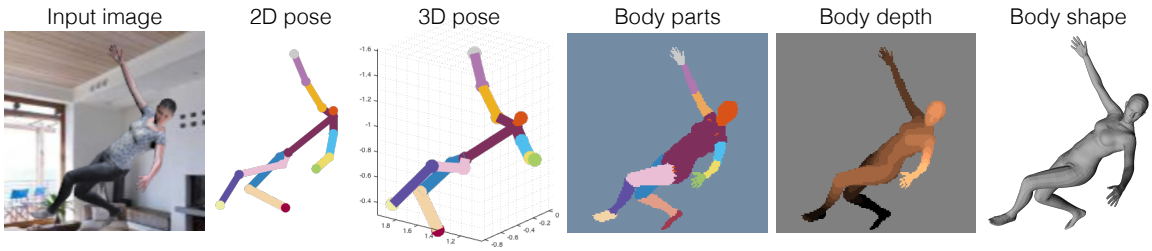
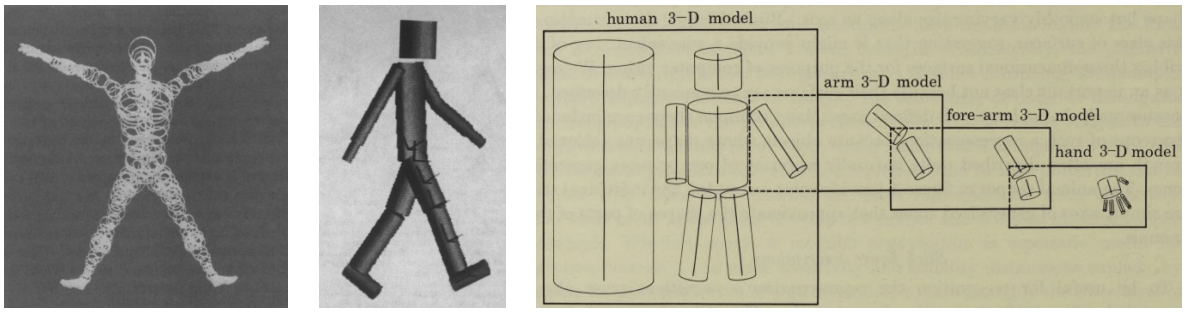


Figure 2-1 – Illustration of computer vision tasks for estimating human body related information.

2.1.1 Articulated pose estimation

Early approaches. Early work in articulated pose estimation used model-based approaches to pose estimation. O'Rourke and Badler [1980] rely on the human body model definition of Badler et al. [1979] which is illustrated in Figure 2-2(a). The model consists of about 600 sphere primitives. It has 25 articulated joints that connect 24 rigid segments. The model incorporates angle limits and collision detection. Similarly, Marr et al. [1978] and Hogg [1983] use a collection of hierarchical 3D cylinders to describe people, see Figure 2-2(b-c). Lee and Chen [1985] recover 3D stick figure from known 2D joint locations (shown in Figure 2-3(a)) by pruning a binary interpretation tree that represents all possible body configurations. Forsyth and Fleck [1997] introduce the concept of *body plans* which defines a sequential grouping of parts for finding people in images based again on cylinder primitives. Researchers then realized the difficulty of recovering articulated 3D models from single images and turned more to 2D pose estimation.

Pictorial structures of Fischler and Elschlager [1973] or puppet-like representations of Hinton [1976] (shown in Figure 2-3(b)) have been widely used later in 2000s. Ronfard et al. [2002] present a part-based approach to parse images of people. They employ a 2D articulated appearance model composed of 15 rectangles as illustrated in Figure 2-3(c). Ioffe and Forsyth [2001] detect people with 2D coarse part-based models. Felzenszwalb and Huttenlocher [2005] describe an efficient algorithm to sample body configurations in the form of pictorial structures given the binary image



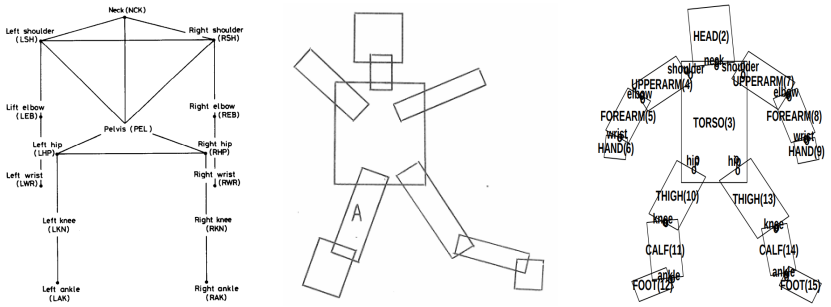
(a) (b) (c)

Figure 2-2 – Early 3D body models using sphere primitives (a) and cylinders (b-c). Figures are taken from [O'Rourke and Badler, 1980; Hogg, 1983; Marr et al., 1978] from left to right.

of the person silhouette. Pictorial structures have been later adopted by the works of [Ramanan et al., 2005; Ramanan, 2006; Ferrari et al., 2008; Eichner et al., 2012; Andriluka et al., 2012].

Prior to the rise of convolutional neural networks (CNNs), the state-of-the-art methods were mainly using *deformable part-models*, which can be considered as an extension to pictorial structures. Deformable part-models were proposed by Felzenszwalb et al. [2008, 2010] and were made more flexible by Yang and Ramanan [2011, 2013]. The latter work defines part mixture models which can capture co-occurrence relations between parts. A parallel line of work has been performed on *poselets* [D. Bourdev and Malik, 2009; Gkioxari et al., 2014]. Poselets are learned clusters of body parts with a data-driven approach. With the success of Random Forests [Amit and Geman, 1997], several methods [Shotton et al., 2011; Charles et al., 2013] followed a regressor-based approach to directly and efficiently output the joint coordinates. An extensive survey on pose estimation can be found in [Moeslund et al., 2013] and more recent works in [Pfister, 2015].

CNN approaches. While earlier work focuses on part detectors and graphical models [Andriluka et al., 2009], recent methods on human pose estimation are based on CNNs [Newell et al., 2016; Pishchulin et al., 2016; Toshev and Szegedy, 2014; Wei



(a) (b) (c)

Figure 2-3 – Early 2D models representing stick figures (a) and pictorial structures (b-c). Figures are taken from [Lee and Chen, 1985; Hinton, 1976; Ronfard et al., 2002] from left to right.

et al., 2016]. Next, we will review 2D and 3D pose estimation approaches separately because the state of the art differs for these two tasks. The availability of large-scale datasets with 2D pose annotation (FLIC [Sapp and Taskar, 2013], LSP [Johnson and Everingham, 2010], MPII [Andriluka et al., 2014], MS-COCO [Lin et al., 2014]) has led to methods which obtain impressive results on 2D pose estimation. The success of 2D pose estimation methods then triggered recent advances in 3D human pose estimation.

• **2D human pose estimation.** Toshev and Szegedy [2014] introduce DeepPose, a cascaded CNN architecture that inputs an image and outputs the xy coordinates of each body joint. Tompson et al. [2014a] present a hybrid method that trains a CNN jointly with a Markov Random Field. Similarly, Chen and Yuille [2014] combine graphical models with CNNs to learn presence of parts and their spatial relationships. Carreira et al. [2016] propose the iterative error feedback approach to iteratively refine the predictions. Instead of predicting coordinates, the community has shifted towards heatmap pose representations as the target formulation in CNNs. Heatmaps are used by [Tompson et al., 2014a; Carreira et al., 2016; Wei et al., 2016; Newell et al., 2016]. For each body joint, a 2D Gaussian with a small mean and variance centered at the ground truth joint location is constructed. The neural network regresses these heatmaps typically with Mean Squared Error (MSE). At test time, the

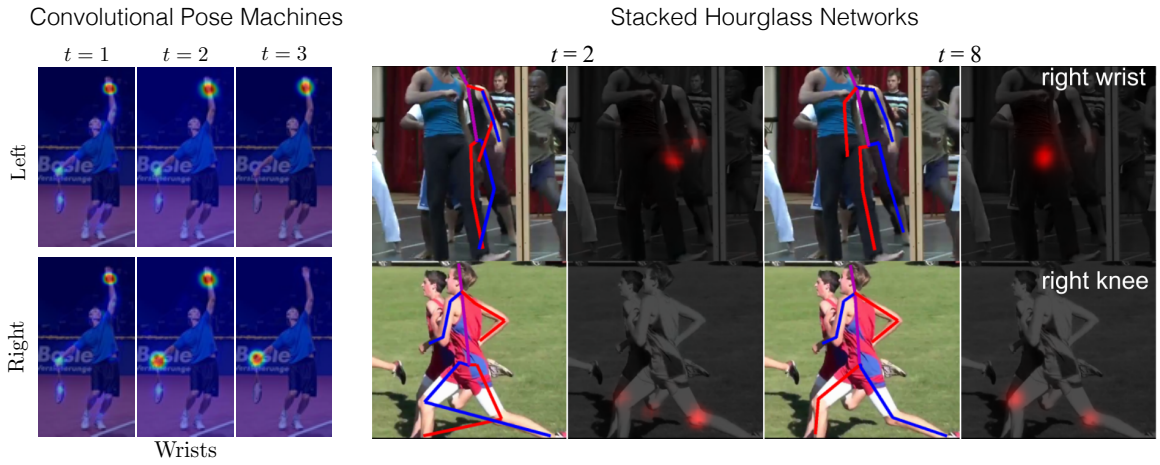


Figure 2-4 – Left/right ambiguities are gradually resolved in the heatmap representations, which initially predict multiple or wrong locations. t denotes the stage (i.e., stack) number. Visualizations are borrowed from convolutional pose machines [Wei et al., 2016] and stacked hourglass networks [Newell et al., 2016].

location with the maximum value becomes the estimated coordinate. Such approach has advantages to consider the uncertainty as well as the multiple hypotheses during training. Compared to the highly non-linear coordinate estimation, heatmaps make the problem more manageable. Convolutional pose machines [Wei et al., 2016] and stacked hourglass networks [Newell et al., 2016] exploit this idea and incorporate intermediate supervision in a cascaded architecture. This helps resolving to a single solution gradually from multiple hypotheses. Figure 2-4 illustrates cases where the first stages (i.e., stack) have difficulty determining left/right limb and the last stage successfully resolves such ambiguity.

More recently, some of these approaches have been extended to support multi-person pose detection [Pishchulin et al., 2016; Insafutdinov et al., 2016; Cao et al., 2017; Newell et al., 2017]. Newell et al. [2017] define associative embedding channels, additional outputs encoding assignments of joints to people. Cao et al. [2017] employ part affinity fields to associate parts of a person. OpenPose software released by Cao et al. [2017] is currently widely used as the state-of-the-art 2D pose estimator.

- **3D human pose estimation.** In the context of 3D pose estimation from a

single image, a common strategy is to lift the given 2D joint estimates to 3D. One approach is to use a database of known 3D poses. 2D pose is treated as a feature to reconstruct the 3D pose either as the nearest sample [Yasin et al., 2016] or a sparse linear combination of known poses in the database [Akhter and Black, 2015; Ramakrishna et al., 2012; Zhou et al., 2016a]. However, the accuracy of such methods highly depend on the size and the variability of the database.

One of the earliest methods using an end-to-end CNN approach is the work of Li and Chan [2014]. They present a multi-task architecture that simultaneously detects 2D body parts and regresses 3D coordinates to directly estimate the 3D pose. Given the success of 2D pose estimation methods, more recent works use neural networks that take 2D joint locations as input and predict the 3D joint coordinates [Martinez et al., 2017; Moreno-Noguer, 2017]. However, 2D joint locations alone are often not sufficient to disambiguate between several 3D poses that have the same 2D projection. Consequently, many approaches have focused on regressing 3D pose directly from features extracted either from a single image [Pavlakos et al., 2017; Zhou et al., 2016b, 2017] or consecutive video frames [Tekin et al., 2016]. Even though 2D pose information alone is ambiguous, it provides strong 3D cues when combined with RGB features. As a result, different approaches have focused on either first predicting 2D pose and combining the extracted features with RGB [Tome et al., 2017] or jointly predicting 2D and 3D pose [Mehta et al., 2017; Popa et al., 2017; Rogez and Schmid, 2016; Rogez et al., 2017].

Regarding the 3D pose representation in neural network frameworks, similar to 2D pose, researchers have started with 3D joint coordinates [Li and Chan, 2014; Martinez et al., 2017]. The target vector is either the metric xyz location of each joint in camera coordinate, root-relative coordinate, or normalized coordinates. The resulting vector has a dimension $J \times 3$, where J is the number of joints. Pavlakos et al. [2017] have extended the 2D heatmap regression to 3D. As shown in Figure 2-5, they defined 3D Gaussians centered at the joint locations in the discretized voxel grid. The xy

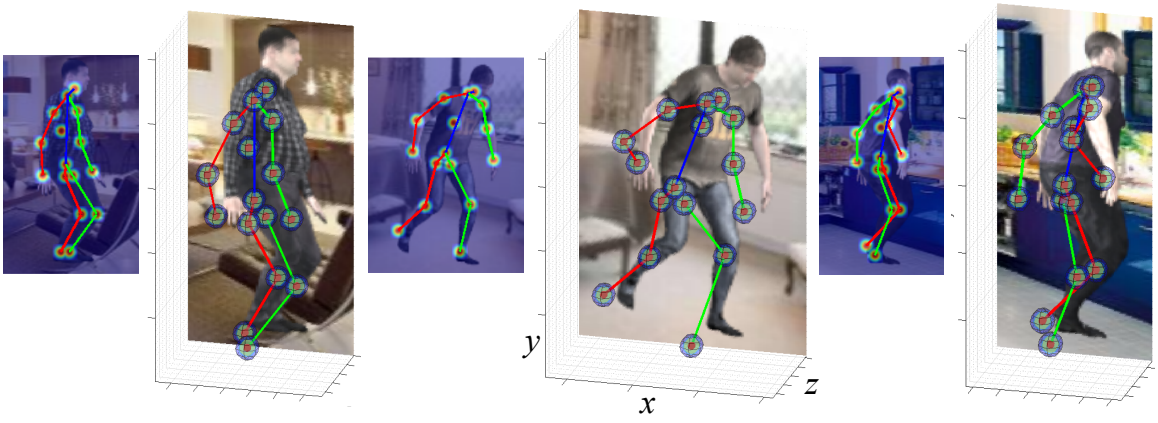


Figure 2-5 – 3D human pose can be represented as 3D heatmaps. Three example images from the SURREAL dataset are visualized with their 2D pose and 3D pose heatmaps. A Gaussian distribution is centered at the joint location. In practice, each body joint has a separate heatmap although heatmaps for all joints are merged for visualization purposes.

coordinates are in the image space. The z coordinate is the discretized metric distance relative to the root joint. In this case, the depth of the root joint and the focal length of the camera are needed to reconstruct the metric coordinates. In Chapter 4, we employ a similar idea to represent 3D pose and we further extend volumetric representation for 3D body shapes. The work of Pavlakos et al. [2018a] also investigates different 3D pose representations and different forms of supervisions such as ordinal depth supervision which confirms the advantages of volumetric representations.

One big unsolved challenge in 3D pose estimation as opposed to 2D pose estimation is the generalization capability as discussed in Section 1.3. Methods trained on constrained motion capture (MoCap) datasets such as HumanEva [Sigal et al., 2010] and Human3.6M [Ionescu et al., 2011, 2014b] usually overfit to these settings. To overcome this issue, the researchers [Kanazawa et al., 2018a; Pavlakos et al., 2018b] have turned to weakly supervised approaches that leverage joint training of *in-the-wild* images and MoCap datasets. We will discuss these methods that employ a human body shape model in Section 2.1.4 in more detail.

2.1.2 Body part segmentation

The semantic segmentation of RGB images is an active area which seeks to parse images into semantic categories such as people, cars, trees etc. Previous works explored Conditional Random Fields (CRFs) [Lucchi et al., 2011; Maire et al., 2011], superpixels [Farabet et al., 2013; Hariharan et al., 2015] and region proposals [Hariharan et al., 2014]. More recently, fully convolutional networks (FCN) [Long et al., 2015] were proposed to allow *end-to-end* training with CNNs. The outputs of these networks have spatial resolution aligned with the input image. DeepLab [Chen et al., 2015a] was proposed to further refine the output of FCNs with CRFs. Ronneberger et al. [2015] introduced the U-Net architecture for biomedical image segmentation. This architecture consists of an encoder and a decoder where the feature maps of the encoder are concatenated to the feature maps at the corresponding decoder layers. It is similar to the stacked hourglass networks [Newell et al., 2016] (discussed in Section 2.1.1) in spirit as they both have symmetric operations between their downsampling and upsampling layers. We will extend the latter architecture for human body part segmentation, 3D body pose estimation, and 3D body shape estimation in Chapters 3 and 4.

In the context of human body part segmentation, impressive results were obtained with RGB-D sensors such as Kinect [Shotton et al., 2011] using random decision forests. They created a synthetic dataset of depth images with ground truth segmentation and 3D pose. Ionescu et al. [2014a] also used random forests, but directly from RGB input. They used the automatically obtained part labels for the Human3.6M MoCap dataset [Ionescu et al., 2014b].

The lack of data has been a problem for studying human body part segmentation from RGB images until recently. Jhuang et al. [2013] annotated a subset of the HMDB51 action dataset [Kuehne et al., 2011] to create the J-HMDB51 dataset. Chen et al. [2014] collected PASCAL-Parts, i.e. part annotations for the PASCAL VOC 2010 dataset [Everingham et al.], and they used only the bounding boxes for a part-

based detection approach on animals. Chen et al. [2016a] used the human body part annotations of [Chen et al., 2014] to train a multi-scale FCN architecture for segmentation. Oliveira et al. [2016] presented a CNN-based approach building on the FCN architecture and collected the Freiburg Sitting People dataset. Zolfaghari et al. [2017] augmented 2D pose annotations of the MPII Human Pose dataset [Andriluka et al., 2014] to create pseudo-ground truth for body part segments. There have been some parallel efforts during the course of this thesis. Lassner et al. [2017] collected the Unite-the-People dataset. Gong et al. [2017] introduced the LIP dataset. We have created the synthetic SURREAL dataset [Varol et al., 2017] (see Chapter 3). Very recently, Güler et al. [2018] collected the DensePose-COCO dataset with manual annotations which allow CNNs trained on this dataset to generalize successfully on challenging images.

2.1.3 Body depth estimation

Depth estimation from a single image has been addressed for generic scenes such as NYUDepth [Silberman et al., 2012] and KITTI [Geiger et al., 2013] datasets [Eigen and Fergus, 2015; Eigen et al., 2014; Liu et al., 2015; Chen et al., 2016c]. Estimating a dense depth map for human bodies is less studied. We present our approach for human depth estimation in Chapter 3 [Varol et al., 2017]. Others have predicted the sparse depth, only for pixels corresponding to body joints [Chen and Ramanan, 2017; Zhou et al., 2017]. Güler et al. [2018] introduced the task of DensePose, which predicts for each pixel its correspondence in a template 3D human body model. Its output can be considered similar to depth due to its 2.5D aspect since they do not deal with the occluded body parts. However, DensePose does not predict metric depth. Next, we present previous works on 3D body shape estimation which, as a side product, can provide the depth estimate once rendered.

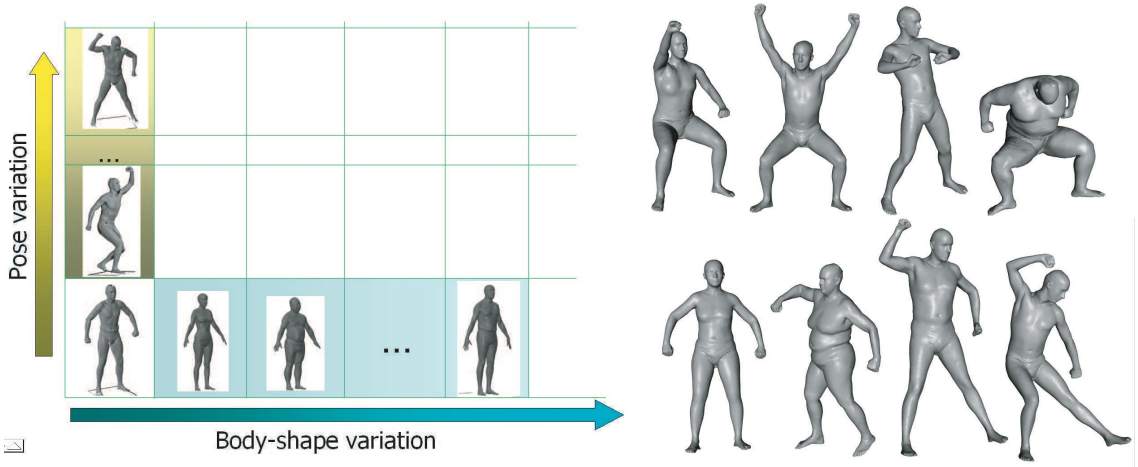
2.1.4 Body shape estimation

We refer to body shape estimation as the recovery of the full surface of the human body. Compared to the skeleton representation which can be recovered from a set of joints, the shape representation is more complicated. Shape recovery from a single image has seen significant advances since the invention of the SCAPE body model [Anguelov et al., 2005]. SCAPE is a statistical model learned from training scans. It is parameterized by *shape* and *pose* parameters separately, which determine triangle deformations. Figure 2-6(a) shows the shape and pose variations captured by the model. Later in 2015, Loper et al. [2015] proposed the SMPL body model which is widely used today. Similar to SCAPE, SMPL decomposes body shape into identity-dependent shape and non-rigid pose-dependent shape. The main difference in SMPL lies in the vertex displacement rather than triangle displacements. Very recently, Pavlakos et al. [2019] extended the SMPL body model by including fully articulated hands and an expressive face, the SMPL-X model. Samples from SMPL and SMPL-X are shown in Figure 2-6 (b) and (c), respectively.

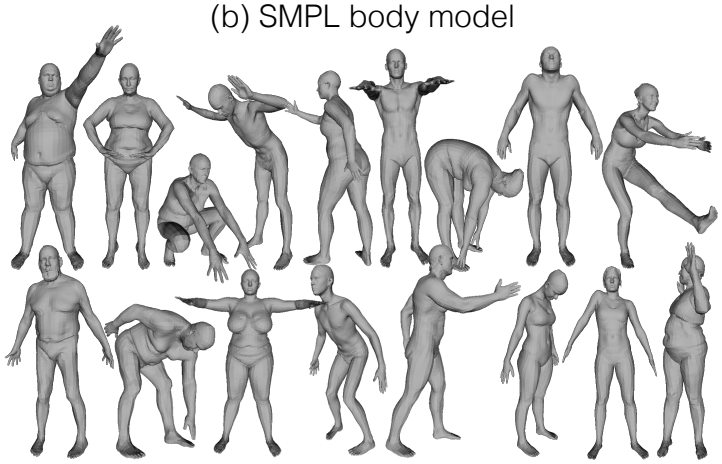
Earlier work [Balan et al., 2007; Sigal et al., 2008; Guan et al., 2009] proposed to optimize pose and shape parameters of the 3D deformable body model SCAPE. The optimization typically aims to fit a model to the image evidence such as silhouettes and 2D joints. To this end, Balan et al. [2007] use silhouettes estimated from multi-camera videos. Sigal et al. [2008] assume true silhouette images, Guan et al. [2009] assume true 2D joints for fitting a model.

More recent methods use the SMPL body model. Given such a model, learned shape and pose *priors*, and an input image, Bogo et al. [2016] present the optimization method, SMPLify, estimating model parameters from a fit to 2D joint estimations. Lassner et al. [2017] extend this approach by incorporating silhouette estimation as additional guidance and improves the optimization performance by densely sampled 2D points. Huang et al. [2017] extend SMPLify for multi-view video sequences with temporal priors. Similar temporal constraints have been used in [Alldieck et al.,

(a) SCAPE body model



(b) SMPL body model



(c) SMPL-X body model

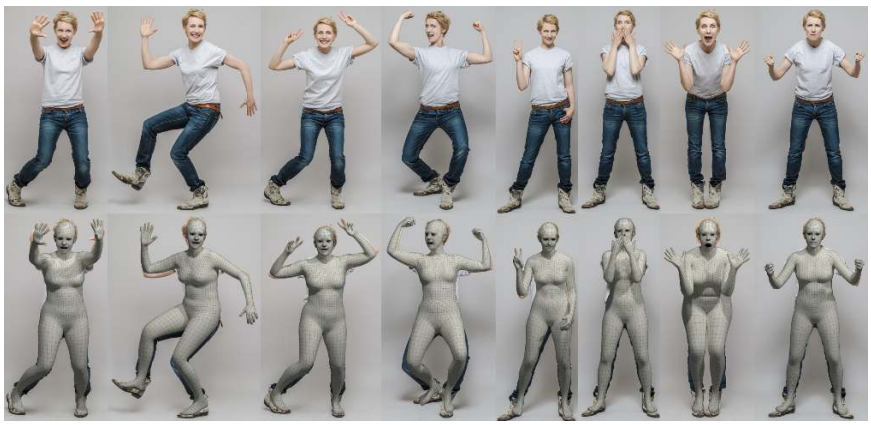


Figure 2-6 – (a): SCAPE body model [Anguelov, 2005] and its decomposition into pose and shape variations. (b): SMPL body model [Loper et al., 2015]. (c): SMPL-X body model [Pavlakos et al., 2019] including fully articulated hands and an expressive face. Figures are taken from [Anguelov, 2005; Loper et al., 2015; Pavlakos et al., 2019].

2017]. Very recently, Pavlakos et al. [2019] incorporate the SMPL-X model in the fitting to 2D detections including hand pose and facial landmarks. These methods are limited by the quality of their 2D detections and depend on priors to regularize the optimization.

Deep neural networks provide an alternative approach that can be expected to learn appropriate priors automatically from the data. Dibra et al. [2016] present one of the first approaches in this direction and train a CNN to estimate the 3D shape parameters from silhouettes, but assume a frontal input view. More recent approaches [Tan et al., 2017; Tung et al., 2017; Kanazawa et al., 2018a] train neural networks to predict the SMPL body parameters from an input image. Tan et al. [2017] design an encoder-decoder architecture that is trained on silhouette prediction and indirectly regresses model parameters at the bottleneck layer. Tung et al. [2017] operate on two consecutive video frames and learn parameters by integrating re-projection loss on the optical flow, silhouettes and 2D joints. Similarly, Kanazawa et al. [2018a] predict parameters with re-projection loss on the 2D joints and introduce an adversary whose goal is to distinguish unrealistic human body shapes. They name their method human mesh recovery (HMR), which implements the linear SMPL model as a differentiable layer in the neural network architecture. This allows the definition of the loss on the 3D joints as well as their 2D projections. Pavlakos et al. [2018b] present a very similar approach that adopts a differentiable SMPL layer. They define the loss on the 3D vertex positions, as well as the 2D silhouette. They further use intermediate predictions such as 2D pose heatmaps and silhouettes to improve generalization from constrained training images. The architectures of [Kanazawa et al., 2018a; Pavlakos et al., 2018b] allow training with weak 2D supervision, which in return allow leveraging in-the-wild images. Omran et al. [2018] extend such approach and use 2D part segmentation as an intermediate representation. In Chapter 4, we will investigate an alternative representation for body shape estimation in a CNN framework. We will use a volumetric representation for both 3D shape (as in Figure 1-

4(d)) and 3D pose (as in Figure 2-5). In Chapter 4, we will rediscuss most relevant works in relation to ours.

We reviewed the literature on human body analysis. Next, we will outline works on human action recognition.

2.2 Human action recognition

In this section, we will review previous research on action recognition given a video input. We will consider the action classification task from dynamic image sequences, and exclude some related tasks such as action classification from static images and spatio-temporal or temporal action localization. We start by an overview of the early works about motion interpretation (Section 2.2.1). Next, we briefly present the era of hand-crafted features (Section 2.2.2). Finally, we outline recent work on learned features using deep neural networks (Section 2.2.3).

2.2.1 Early days in motion interpretation

We refer the reader to [Cédras and Shah, 1995; Shah and Jain, 1997] and [Aggarwal and Cai, 1999; Gavrila, 1999] for detailed surveys covering works up to the end of 1990s on motion-based recognition, and human motion analysis, respectively. Here, we briefly review some of the key works. Cédras and Shah [1995] reviews the literature on *motion verb recognition*, which is similar to what we typically refer to as *action recognition*, i.e. association of natural language verbs with the motion performed by a moving object in a sequence of images[Cédras and Shah, 1995]. Initial works in this direction were presented by [Badler, 1975; Neumann and Novak, 1983]. Figure 2-7 shows the concept hierarchy defined by [Badler, 1975] for predicting motion verbs. They extend the vocabulary of motion verbs gathered by [Miller, 1972], such as *absorbs, accelerates, accompanies* etc.

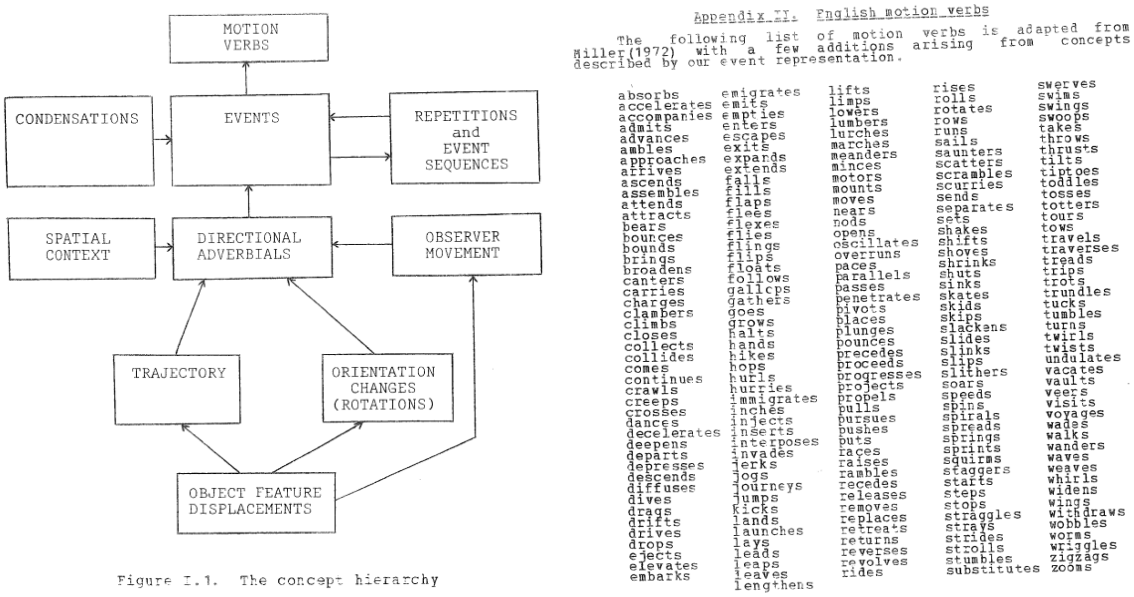


Figure 2-7 – The concept hierarchy of Badler [1975] for predicting motion verbs and their vocabulary of motion verbs extended from [Miller, 1972]. Screenshots are taken from [Badler, 1975].

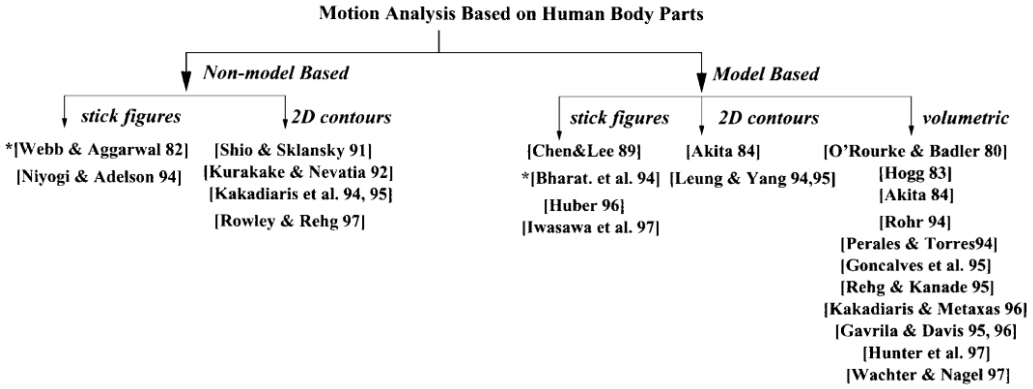


Figure 2-8 – The grouping of early works in motion analysis based on human body parts. We notice the relatively higher number of methods based on 3D models. The diagram is taken from [Aggarwal and Cai, 1999]. We refer to [Aggarwal and Cai, 1999] for the citations.

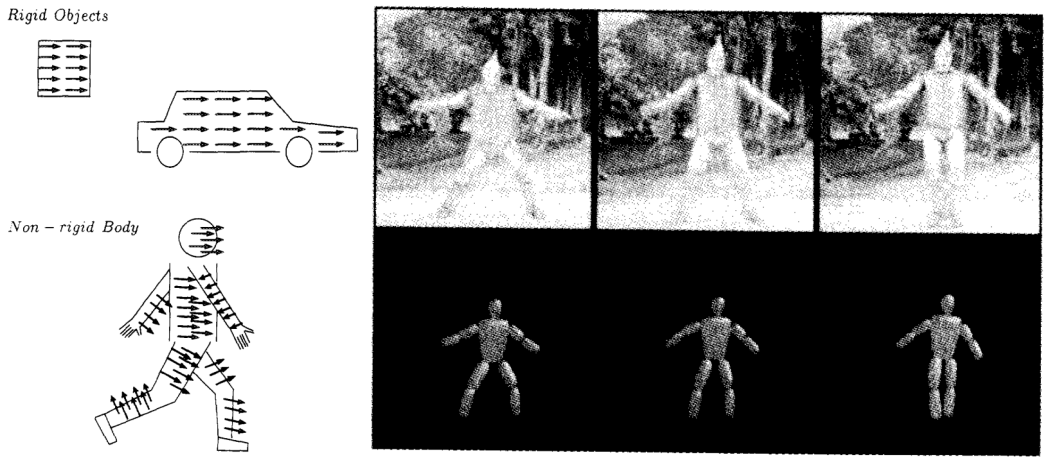


Figure 2-9 – (left) Illustration of rigid object motion versus nonrigid body motion by Rohr [1997]. (right) The tracking results of Pentland and Horowitz [1991] on a jumping sequence by fitting an articulated model to noisy optical flow data. Figures are borrowed from the corresponding papers.

Aggarwal and Cai [1999] group the early works into categories in terms of model-based and non-model-based as depicted in Figure 2-8. We see that early methods for human motion analysis mostly employed model-based approaches [Rohr, 1994; Gavrila and Davis, 1995; Wachter and Nagel, 1997; Bregler and Malik, 1998]. Gavrila and Davis [1995] performed recognition by tracking 3D body models. Similarly, Rohr [1994] and Wachter and Nagel [1997] studied person tracking using models based on [Marr et al., 1978] described in Section 2.1.2.

Later works continued model-based approaches for human tracking. Probabilistic search algorithms were explored to model the distribution over poses [Deutscher et al., 2000; Sidenbladh et al., 2000; Sidenbladh and Black, 2001; Sminchisescu and Triggs, 2003]. Deutscher et al. [2000] applied an annealed particle filtering method to recover body motion. Similarly, Sidenbladh et al. [2000]; Sidenbladh and Black [2001] relied on particle filtering to estimate the parameters of a body model based on cylinders and spheres.

Baumberg and Hogg [1996] used contour-based models to track walking pedestrians. In their earlier work [Baumberg and Hogg, 1994], they made use of active shape

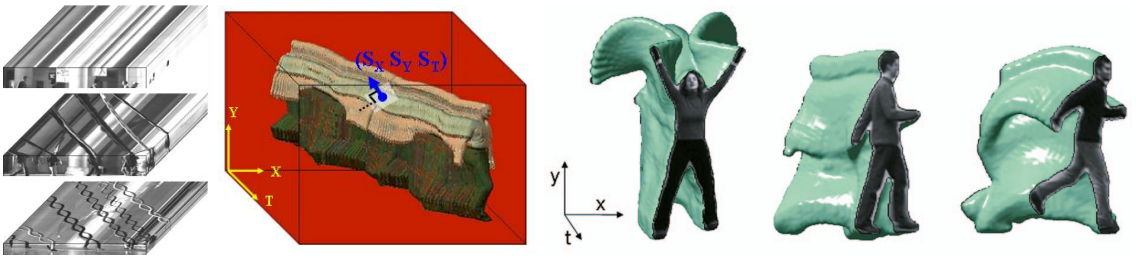


Figure 2-10 – Space-time video representations. (left) The xt slices of the space-time volume xyt at different y values provide gait patterns of people. (middle) Space-time volume of a person walking from left-to-right. (right) Space-time silhouette shapes of jumping-jack, walk, and run actions. Figures are taken from [Niyogi and Adelson, 1994; Zelnik-Manor and Irani, 2001; Gorelick et al., 2007] from left to right.

models and Kalman filter for efficient tracking.

Other works used tracking of image features [Bregler, 1997; Isard and Blake, 1998; Song et al., 2003] for recognizing actions. Bregler [1997] used Hidden Markov models for gait recognition. [Isard and Blake, 1998] addressed hand gesture recognition using particle filtering for feature tracking.

Alternatively, optical flow was used for matching motion templates [Black et al., 1997; Yacoob and Black, 1998; Efros et al., 2003]. Figure 2-9(left) illustrates the complexity of articulated human motion relative to rigid objects. Pentland and Horowitz [1991] proposed fitting of a 3D articulated model to optical flow data and obtained impressive recovery of nonrigid motion shown in Figure 2-9(right). Ju et al. [1996] introduced the notion of cardboard people, a 2D model-based approach. They track the articulated motion using parameterized models of optical flow. Black et al. [1997] used PCA to learn a set of basis flow fields for estimating the motion of human legs while walking. In their later work, Yacoob and Black [1998] extended this approach to action recognition. They collect a dataset of four activities: walking, marching, line-walking, and kicking while walking. Efros et al. [2003] addressed action recognition with particular focus on recordings at a distance. They collect videos from ballet, tennis, and football shootings. They design a spatio-temporal motion descriptor based on optical flow to query nearest neighbor for classification.

On the other hand, Zelnik-Manor and Irani [2001] proposed non-parametric spatio-temporal features based on gradients. Spatio-temporal representations for video had also been used by Niyogi and Adelson [1994] to analyze gait patterns. Zelnik-Manor and Irani [2001] used features from multiple temporal scales to cluster long video sequences into events. Later Gorelick et al. [2007] extended this method to extract a variety of features from the space-time volume. Such descriptors had the advantage to handle videos of different lengths and had the potential to capture long-term dynamics. Figure 2-10 visualizes the space-time characteristics of several actions. We will discuss our approach to *learn* hierarchical space-time features with convolutional neural networks in Chapter 5. In the next section, we review more recent methods based on hand-crafted video features.

2.2.2 Hand-crafted video features

Action recognition in the last decade has been dominated by local video features [Schüldt et al., 2004; Laptev et al., 2008; Wang and Schmid, 2013] aggregated with Bag-of-Features histograms [Csurka et al., 2004] or Fisher Vector representations [Perronnin et al., 2010]. Such pipelines consist of feature sampling, description, aggregation, and classification steps. Laptev [2005] extended the gradient-based SIFT [Lowe, 2004] image feature detector and introduced space-time interest point (STIP) detectors for sampling video features. Wang et al. [2013a] showed that descriptor sampling along dense trajectories outperforms sparse sampling. In a follow up study, Wang and Schmid [2013] introduced improved dense trajectories (IDT) by camera motion calibration, which further improved recognition performance in unconstrained videos. Today, IDT is still a competitive approach compared to current CNN-based state-of-the-art methods.

There is a vast literature on hand-crafting descriptors for action recognition. Some of the popular low-level descriptors are histograms of 3D gradient orientations (HOG3D) [Scovanner et al., 2007; Kläser et al., 2008], oriented histograms of

flow (HOF) [Laptev et al., 2008], and motion boundary histograms (MBH) [Dalal et al., 2006; Wang et al., 2013a]. Action banks [Sadanand and Corso, 2012], action attributes [Liu et al., 2011a], actons [Zhu et al., 2013], and motionlets [Wang et al., 2013b] are proposed as mid-level features.

While traditional pipelines resemble earlier methods for object recognition, the use of local motion features, in particular Motion Boundary Histograms [Dalal et al., 2006; Wang et al., 2013a], has been found important for action recognition in practice. Explicit representations of the temporal structure of actions have rarely been used with some exceptions such as the recent work [Fernando et al., 2015]. Next, we review recent efforts on learning action recognition representations with neural networks.

2.2.3 Learned video features

With the availability of large-scale image datasets [Deng et al., 2009; Russakovsky et al., 2015], deep neural networks have quickly taken over the majority of still-image recognition tasks such as object recognition [Krizhevsky et al., 2012]. The gain for action recognition was not immediate despite the active efforts spent on exploring a wide range of architectures suitable for video inputs.

The architectures can be grouped in three categories: (i) image-based 2D convolutional neural networks, (ii) recurrent neural networks (RNNs), and (iii) space-time 3D convolutional neural networks. The first straightforward extension of image-based CNNs to videos is to treat multi-frame image sequences as multi-channel images [Karpathy et al., 2014; Simonyan and Zisserman, 2014; Wang et al., 2015b, 2016a]. This approach does not explicitly model the temporal structure of actions. Second, the temporal aspect of videos naturally led to the exploration of RNNs, mainly long-short term memory networks (LSTMs) [Hochreiter and Schmidhuber, 1997; Donahue et al., 2015; Ng et al., 2015]. However, there were no significant gains obtained by using RNNs over 2D CNNs. Third, 3D CNNs have been investigated on the spatio-temporal video inputs [Ji et al., 2010; Taylor et al., 2010; Tran et al., 2015]. Compared

to the RNNs operating on top of the last-layer CNN features, 3D CNNs are able to hierarchically capture local spatio-temporal features. However, 3D CNNs are often found computationally expensive. Alternative architectures have been proposed by Sun et al. [2015a] who factorize spatial and temporal convolutions, and by Ballas et al. [2016] who introduce a hybrid gated recurrent unit (GRU) architecture.

Among the aforementioned approaches, variants of two-stream architectures have been the most used for certain period [Simonyan and Zisserman, 2014; Wang et al., 2015a,b, 2016a; Bilen et al., 2016; Feichtenhofer et al., 2016]. The two-stream approach typically combines the static RGB image stream with the pre-computed stacked optical flow stream. There have also been multi-stream attempts by incorporating pose features as input [Zolfaghari et al., 2017]. On the other hand, Ng et al. [2018] introduced ActionFlowNet, a multi-task learning scheme for optical flow and action classification directly on raw RGB data.

More recently, two-stream 3D CNNs have shown successful performance. We present one such approach [Varol et al., 2018b] in this thesis (Chapter 5). More recent work [Carreira and Zisserman, 2017] demonstrates excellent performance when trained on the new Kinetics dataset [Kay et al., 2017]. Carreira and Zisserman [2017] present I3D, inflated 3D CNNs, whose weights are initialized by copying ImageNet pre-trained 2D CNN weights across time. Similar initialization strategies were also investigated in [Mansimov et al., 2015]. Recently, non-local neural networks were proposed by Wang et al. [2018c] as a generic component for capturing long-range dependencies. For further architecture search studies, we refer the reader to [Tran et al., 2018; Hara et al., 2018].

In Chapters 5 and 6, we will present our work on action recognition focused on long-term temporal convolutions and on view-independent video representations, respectively. In these chapters, we will rediscuss most relevant prior work in relation to ours.

Part I

Human Body Analysis

Chapter 3

Learning from Synthetic Humans

This chapter presents our first contribution to human body analysis, with the focus on the generation of synthetic person images and using this data for training visual models.

Estimating human pose, shape, and motion from images and videos are fundamental challenges with many applications. Recent advances in 2D human pose estimation use large amounts of manually-labeled training data for learning convolutional neural networks (CNNs). Such data is time consuming to acquire and difficult to extend. Moreover, manual labeling of 3D pose, depth and motion is impractical. In this chapter, we present SURREAL (Synthetic hUmans foR REAL tasks): a new large-scale dataset with synthetically-generated but realistic images of people rendered from 3D sequences of human motion capture data. We generate more than 6 million frames together with ground truth pose, depth maps, and segmentation masks. We show that CNNs trained on our synthetic dataset allow for accurate human depth estimation and human part segmentation in real RGB images. Our results and the new dataset open up new possibilities for advancing person analysis using automatically-generated and large-scale synthetic data.

3.1 Introduction

Convolutional Neural Networks provide significant gains to problems with large amounts of training data. In the field of human analysis, recent datasets [Andriluka et al., 2014; Sapp and Taskar, 2013] now gather a sufficient number of annotated images to train networks for 2D human pose estimation [Newell et al., 2016; Wei et al., 2016]. Other tasks such as accurate estimation of human motion, depth and body-part segmentation are lagging behind as manual supervision for such problems at large scale is prohibitively expensive.

Images of people have rich variation in poses, clothing, hair styles, body shapes, occlusions, viewpoints, motion blur and other factors. Many of these variations, however, can be synthesized using existing 3D motion capture (MoCap) data [Carnegie-Mellon Mocap Database; Ionescu et al., 2014b] and modern tools for realistic rendering. Provided sufficient realism, such an approach would be highly useful for many tasks as it can generate rich ground truth in terms of depth, motion, body-part segmentation and occlusions.

Although synthetic data has been used for many years, realism has been limited. In this work we present SURREAL: a new large-scale dataset with synthetically-generated but realistic images of people. Images are rendered from 3D sequences of MoCap data. To ensure realism, the synthetic bodies are created using the SMPL body model [Loper et al., 2015], whose parameters are fit by the MoSh [Loper et al., 2014] method given raw 3D MoCap marker data. We randomly sample a large variety of viewpoints, clothing and lighting. SURREAL contains more than 6 million frames together with ground truth pose, depth maps, and segmentation masks. We show that CNNs trained on synthetic data allow for accurate human depth estimation and human part segmentation in real RGB images, see Figure 3-1. Here, we demonstrate that our dataset, while being synthetic, reaches the level of realism necessary to support training for multiple complex tasks. This opens up opportunities for training deep networks using graphics techniques available now. SURREAL dataset is publicly

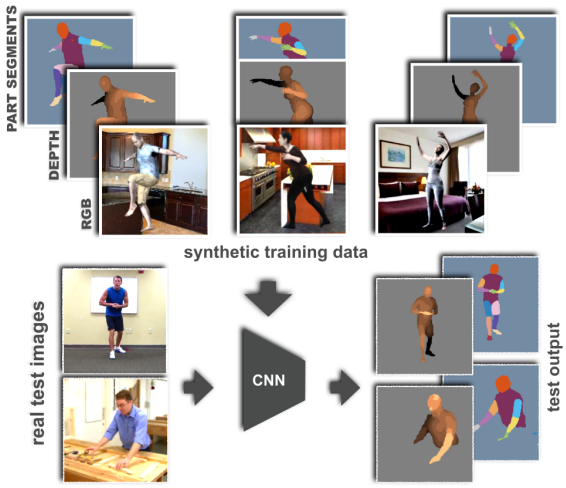


Figure 3-1 – We generate photo-realistic synthetic images and their corresponding ground truth for learning pixel-wise classification problems: human part segmentation and depth estimation. The convolutional neural network trained only on synthetic data generalizes to real images sufficiently for both tasks. Real test images in this figure are taken from MPII Human Pose dataset [Andriluka et al., 2014].

available together with the code to generate synthetic data and to train models for body part segmentation and depth estimation [SURREAL project page].

The rest of this chapter is organized as follows. Section 3.2 reviews related work. Section 3.3 presents our approach for generating realistic synthetic videos of people. In Section 3.4 we describe our CNN architecture for human body part segmentation and depth estimation. Section 3.5 reports experiments. We conclude in Section 3.6.

3.2 Related work

Knowledge transfer from synthetic to real images has been recently studied with deep neural networks. Dosovitskiy et al. [2015] learn a CNN for optical flow estimation using synthetically generated images of rendered 3D moving chairs. Peng et al. [2015] study the effect of different visual cues such as object/background texture and color when rendering synthetic 3D objects for object detection task. Similarly, Su et al. [2015] explores rendering 3D objects to perform viewpoint estimation. Fanello et al.

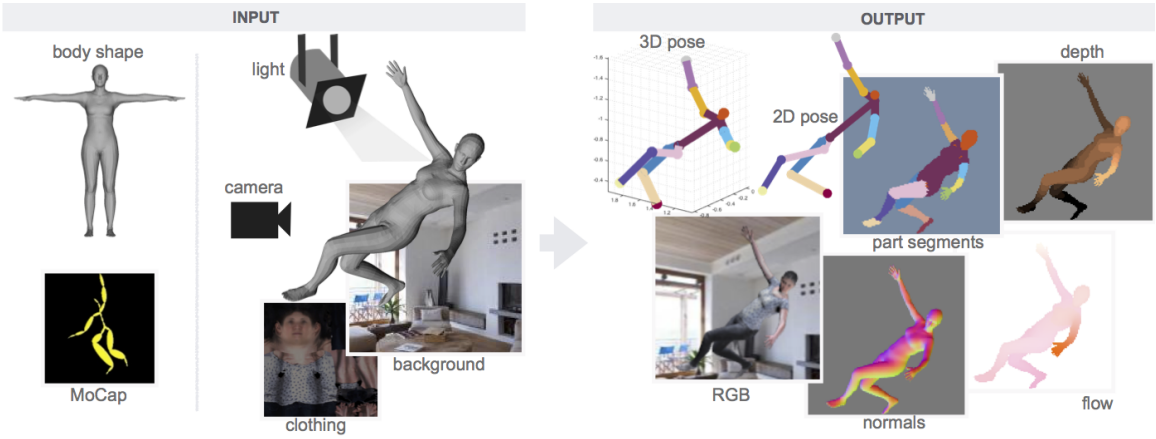


Figure 3-2 – Our pipeline for generating synthetic data. A 3D human body model is posed using motion capture data and a frame is rendered using a background image, a texture map on the body, lighting and a camera position. These ingredients are randomly sampled to increase the diversity of the data. We generate RGB images together with 2D/3D poses, surface normals, optical flow, depth images, and body-part segmentation maps for rendered people.

[2014] render synthetic infrared images of hands and faces to predict depth and parts. Recently, Gaidon et al. [2016] have released the Virtual KITTI dataset with synthetically generated videos of cars to study multi-object tracking.

Several works focused on creating synthetic images of human bodies for learning 2D pose estimation [Pishchulin et al., 2012; Qiu, 2016; Romero et al., 2015], 3D pose estimation [Chen et al., 2016b; Du et al., 2016; Ghezelghieh et al., 2016; Okada and Soatto, 2008; Rogez and Schmid, 2016; Sminchisescu et al., 2006; Zhou et al., 2016a], pedestrian detection [Marin et al., 2010; Pishchulin et al., 2012, 2011], and action recognition [Rahmani and Mian, 2015, 2016]. Pishchulin et al. [2011] generate synthetic images with a game engine. Pishchulin et al. [2012] deform 2D images with a 3D model. More recently, Rogez and Schmid [2016] use an image-based synthesis engine to augment existing real images. Ghezelghieh et al. [2016] render synthetic images with 10 simple body models with an emphasis on upright people; however, the main challenge using existing MoCap data for training is to generalize to poses that are not upright. Human3.6M dataset [Ionescu et al., 2014b] presents realistic

rendering of people in mixed reality settings; however, the approach to create these is expensive.

A similar direction has been explored in [Rahmani and Mian, 2015, 2016; Rhodin et al., 2016; Shotton et al., 2011]. In [Rahmani and Mian, 2015], action recognition is addressed with synthetic human trajectories from MoCap data. [Rahmani and Mian, 2016; Shotton et al., 2011] train CNNs with synthetic depth images. EgoCap [Rhodin et al., 2016] creates a dataset by augmenting egocentric sequences with background.

The closest approach to this work is [Chen et al., 2016b], where the authors render large-scale synthetic images for predicting 3D pose with CNNs. Our dataset differs from [Chen et al., 2016b] by having a richer, per-pixel ground truth, thus allowing to train for pixel-wise predictions and multi-task scenarios. In addition, we argue that the realism in our synthetic images is better (see sample videos in SURREAL project page), thus resulting in a smaller gap between features learned from synthetic and real images. The method in [Chen et al., 2016b] heavily relies on real images as input in their training with domain adaptation. This is not the case for our synthetic training. Moreover, we render video sequences which can be used for temporal modeling.

Our dataset presents several differences with existing synthetic datasets. It is the first large-scale person dataset providing depth, part segmentation and flow ground truth for synthetic RGB frames. Other existing datasets are used either for taking RGB image as input and training only for 2D/3D pose, or for taking depth/infrared images as input and training for depth/parts segmentation. In this chapter, we show that photo-realistic renderings of people under large variations in shape, texture, viewpoint and pose can help solving pixel-wise human labeling tasks.

3.3 Data generation

This section presents our SURREAL (Synthetic hUmans foR REAL tasks) dataset and describes key steps for its generation (Section 3.3.1). We also describe how we

Figure 3-3 – Sample frames from our SURREAL dataset with a large variety of poses, body shapes, clothings, viewpoints and backgrounds.

obtain ground truth data for real MoCap sequences (Section 3.3.2).

3.3.1 Synthetic humans

Our pipeline for generating synthetic data is illustrated in Figure 3-2. A human body with a *random* 3D pose, *random* shape and *random* texture is rendered from a *random* view-point for some *random* lighting and a *random* background image. Below we define what "random" means in all these cases. Since the data is synthetic, we also generate ground truth depth maps, optical flow, surface normals, human part segmentations and joint locations (both 2D and 3D). As a result, we obtain 6.5 million frames grouped into $67,582$ continuous image sequences. See Table 3.1 for more statistics, Section 3.5.2 for the description of the synthetic train/test split, and Figure 3-3 for samples from the SURREAL dataset.

Body model. Synthetic bodies are created using the SMPL body model [Loper et al., 2015]. SMPL is a realistic articulated model of the body created from thousands of high-quality 3D scans, which decomposes body deformations into pose (kinematic deformations due to skeletal posture) and shape (body deformations intrinsic to a particular person that make them different from others). SMPL is compatible with most animation packages like Blender [Blender - a 3D modelling and rendering package].

SMPL deformations are modeled as a combination of linear blend skinning and linear blendshapes defined by principal components of body shape variation. SMPL pose and shape parameters are converted to a triangulated mesh using Blender, which then applies texture, shading and adds a background to generate the final RGB output.

Body shape. In order to render varied, but realistic, body shapes we make use of the CAESAR dataset [Robinette et al., 2002], which was used to train SMPL. To create a body shape, we select one of the CAESAR subjects at random and approximate their shape with the first 10 SMPL shape principal components. Ten shape components explain more than 95% of the shape variance in CAESAR (at the resolution of our mesh) and produce quite realistic body shapes.

Body pose. To generate images of people in realistic poses, we take motion capture data from the CMU MoCap database [Carnegie-Mellon Mocap Database]. CMU MoCap contains more than 2000 sequences of 23 high-level action categories, resulting in more than 10 hours of recorded 3D locations of body markers.

It is often challenging to realistically and automatically retarget MoCap skeleton data to a new model. For this reason we do not use the skeleton data but rather use MoSh [Loper et al., 2014] to fit the SMPL parameters that best explain raw 3D MoCap marker locations. This gives both the 3D shape of the subject and the articulated pose parameters of SMPL. To increase the diversity, we replace the estimated 3D body shape with a set of randomly sampled body shapes.

We render each CMU MoCap sequence three times using different random parameters. Moreover, we divide the sequences into clips of 100 frames with 30%, 50% and 70% overlaps for these three renderings. Every pose of the sequence is rendered with consistent parameters (i.e. body shape, clothing, light, background etc.) within each clip.

Human texture. We use two types of real scans for the texture of body models. First, we extract SMPL texture maps from CAESAR scans, which come with a color texture per 3D point. These maps vary in skin color and person identities, however, their quality is often low due to the low resolution, uniform tight-fitting clothing, and visible markers placed on the face and the body. Anthropometric markers are automatically removed from the texture images and inpainted. To provide more variety, we extract a second set of textures obtained from 3D scans of subjects with normal clothing. These scans are registered with 4Cap as in [Pons-Moll et al., 2015]. The texture of real clothing substantially increases the realism of generated images, even though SMPL does not model 3D deformations of clothes.

20% of our data is rendered with the first set (158 CAESAR textures randomly sampled from 4000), and the rest with the second set (772 clothed textures). To preserve the anonymity of subjects, we replace all faces in the texture maps by the average CAESAR face. The skin color of this average face is corrected to fit the face skin color of the original texture map. This corrected average face is blended smoothly with the original map, resulting in a realistic and anonymized body texture.

Light. The body is illuminated using Spherical Harmonics with 9 coefficients [Green, 2003]. The coefficients are randomly sampled from a uniform distribution between -0.7 and 0.7, apart from the ambient illumination coefficient (which has a minimum value of 0.5) and the vertical illumination component, which is biased to encourage the illumination from above. Since Blender does not provide Spherical Harmonics illumination, a spherical harmonic shader for the body material was implemented in Open Shading Language.

Camera. The projective camera has a resolution of 320×240, focal length of 60mm and sensor size of 32mm. To generate images of the body in a wide range of positions, we take 100-frame MoCap sub-sequences and, in the first frame, render the body so that the center of the viewport points to the pelvis of the body, at a random distance

(sampled from a normal distribution with 8 meters mean, 1 meter deviation) with a random yaw angle. The remainder of the sequence then effectively produces bodies in a range of locations relative to the static camera.

Background. We render the person on top of a static background image. To ensure that the backgrounds are reasonably realistic and do not include other people, we sample from a subset of LSUN dataset [Yu et al., 2015] that includes total of 400K images from the categories kitchen, living room, bedroom and dining room.

Ground truth. We perform multiple rendering passes in Blender to generate different types of per-pixel ground truth. The *material* pass generates pixel-wise segmentation of rendered body parts, given different material indices assigned to different parts of our body model. The *velocity* pass, typically used to simulate motion blur, provides us with a render simulating optical flow. The *depth* and *normal* passes, used for emulating effects like fog, bokeh or for performing shading, produce per-pixel depth maps and normal maps. The final texture rendering pass overlays the shaded, textured body over the random background. Together with this data we save camera and lighting parameters as well as the 2D/3D positions of body joints.

3.3.2 Generating ground truth for real human data

Human3.6M dataset [Ionescu et al., 2011, 2014b] provides ground truth for 2D and 3D human poses. Additionally, a subset of the dataset (H80K) [Ionescu et al., 2014a] has segmentation annotation, but the definition of parts is different from the SMPL body parts used for our training. We complement this ground truth and generate predicted SMPL body-part segmentation and depth maps for people in Human3.6M for all frames. Here again we use MoSh [Loper et al., 2014] to fit the SMPL body shape and pose to the raw MoCap marker data. This provides a good fit of the model to the shape and the pose of real bodies. Given the provided camera calibration, we

Table 3.1 – SURREAL dataset in numbers. Each MoCap sequence is rendered 3 times (with 3 different overlap ratios). Clips are mostly 100 frames long. We obtain a total of 6,5 million frames.

	#subjects	#sequences	#clips	#frames
Train	115	1,964	55,001	5,342,090
Test	30	703	12,528	1,194,662
Total	145	2,607	67,582	**6,536,752**

project models to images. We then render the ground truth segmentation, depth, and 2D/3D joints as above, while ensuring correspondence with real pixel values in the dataset. The depth is different from the time-of-flight (depth) data provided by the official dataset. These MoSh fits provide a form of approximate "ground truth". See Figures 3-6 and 3-7 for generated examples. We use this for evaluation on the test set as well as for the baseline where we train only on real data, and also for fine-tuning our models pre-trained on synthetic data. In the rest of the chapter, all frames from the synthetic training set are used for synthetic pre-training.

3.4 Approach

In this section, we present our approach for human body part segmentation [Chen et al., 2016a; Oliveira et al., 2016] and human depth estimation [Eigen and Fergus, 2015; Eigen et al., 2014; Liu et al., 2015], which we train with synthetic and/or real data, see Section 3.5 for the evaluation.

Our approach builds on the stacked hourglass network architecture introduced originally for 2D pose estimation problem [Newell et al., 2016]. This network involves several repetitions of contraction followed by expansion layers which have skip connections to implicitly model spatial relations from different resolutions that allows bottom-up and top-down structured prediction. The convolutional layers with residual connections and 8 'hourglass' modules are stacked on top of each other, each

successive stack taking the previous stack's prediction as input. The reader is referred to [Newell et al., 2016] for more details. A variant of this network has been used for scene depth estimation [Chen et al., 2016c]. We choose this architecture because it can infer pixel-wise output by taking into account human body structure.

Our network input is a 3-channel RGB image of size 256×256 cropped and scaled to fit a human bounding box using the ground truth. The network output for each stack has dimensions $64 \times 64 \times 15$ in the case of segmentation (14 classes plus the background) and $64 \times 64 \times 20$ for depth (19 depth classes plus the background). We use cross-entropy loss defined on all pixels for both segmentation and depth. The final loss of the network is the sum over 8 stacks. We train for 50K iterations for synthetic pre-training using the RMSprop algorithm with mini-batches of size 6 and a learning rate of 10^{-3}. Our data augmentation during training includes random rotations, scaling and color jittering.

We formulate the problem as pixel-wise classification task for both segmentation and depth. When addressing segmentation, each pixel is assigned to one of the pre-defined 14 human parts, namely head, torso, upper legs, lower legs, upper arms, lower arms, hands, feet (separately for right and left) or to the background class. Regarding the depth, we align ground-truth depth maps on the z-axis by the depth of the pelvis joint, and then quantize depth values into 19 bins (9 behind and 9 in front of the pelvis). We set the quantization constant to 45mm to roughly cover the depth extent of common human poses. The network is trained to classify each pixel into one of the 19 depth bins or background. At test time, we first upsample feature maps of each class with bilinear interpolation by a factor of 4 to output the original resolution. Then, each pixel is assigned to the class for which the corresponding channel has the maximum activation.

3.5 Experiments

We test our approach on several datasets. First, we evaluate the segmentation and depth estimation on the test set of our synthetic SURREAL dataset. Second, we test the performance of segmentation on real images from the Freiburg Sitting People dataset [Oliveira et al., 2016]. Next, we evaluate segmentation and depth estimation on real videos from the Human3.6M dataset [Ionescu et al., 2011, 2014b] with available 3D information. Then, we qualitatively evaluate our approach on the more challenging MPII Human Pose dataset [Andriluka et al., 2014]. Finally, we experiment and discuss design choices of the SURREAL dataset.

3.5.1 Evaluation measures

We use intersection over union (IOU) and pixel accuracy measures for evaluating the segmentation approach. The final measure is the average over 14 human parts as in [Oliveira et al., 2016]. Depth estimation is formulated as a classification problem, but we take into account the continuity when we evaluate. We compute root-mean-squared-error (RMSE) between the predicted quantized depth value (class) and the ground truth quantized depth on the human pixels. To interpret the error in real world coordinates, we multiply it by the quantization constant (45mm). We also report a scale and translation invariant RMSE (st-RMSE) by solving for the best translation and scaling in z-axis to fit the prediction to the ground truth. Since inferring depth from RGB is ambiguous, this is a common technique used in evaluations [Eigen et al., 2014].

3.5.2 Validation on synthetic images

Train/test split. To evaluate our methods on synthetic images, we separate 20% of the synthetic frames for the test set and train all our networks on the remaining training set. The split is constructed such that a given CMU MoCap subject is

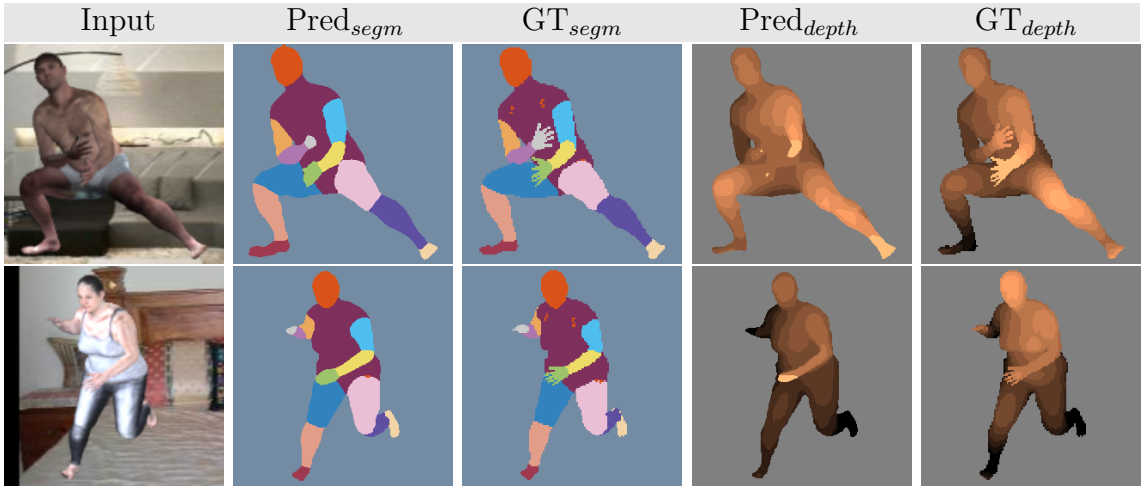


Figure 3-4 – Segmentation and depth predictions on synthetic test set.

assigned as either train or test. Whereas some subjects have a large number of instances, some subjects have unique actions, and some actions are very common (walk, run, jump). Overall, 30 subjects out of 145 are assigned as test. 28 test subjects cover all common actions, and 2 have unique actions. Remaining subjects are used for training. Although our synthetic images have different body shape and appearance than the subject in the originating MoCap sequence, we still found it appropriate to split by subjects. We separate a subset of our body shapes, clothing and background images for the test set. This ensures that our tests are unbiased with regards to appearance, yet are still representative of all actions. Table 3.1 summarizes the number of frames, clips and MoCap sequences in each split. Clips are the continuous 100-frame sequences where we have the same random body shape, background, clothing, camera and lighting. A new random set is picked at every clip. Note that a few sequences have less than 100 frames.

Results on synthetic test set. The evaluation is performed on the middle frame of each 100-frame clip on the aforementioned held-out synthetic test set, totaling in 12,528 images. For segmentation, the IOU and pixel accuracy are 69.13% and 80.61%, respectively. Evaluation of depth estimation gives 72.9mm and 56.3mm for RMSE

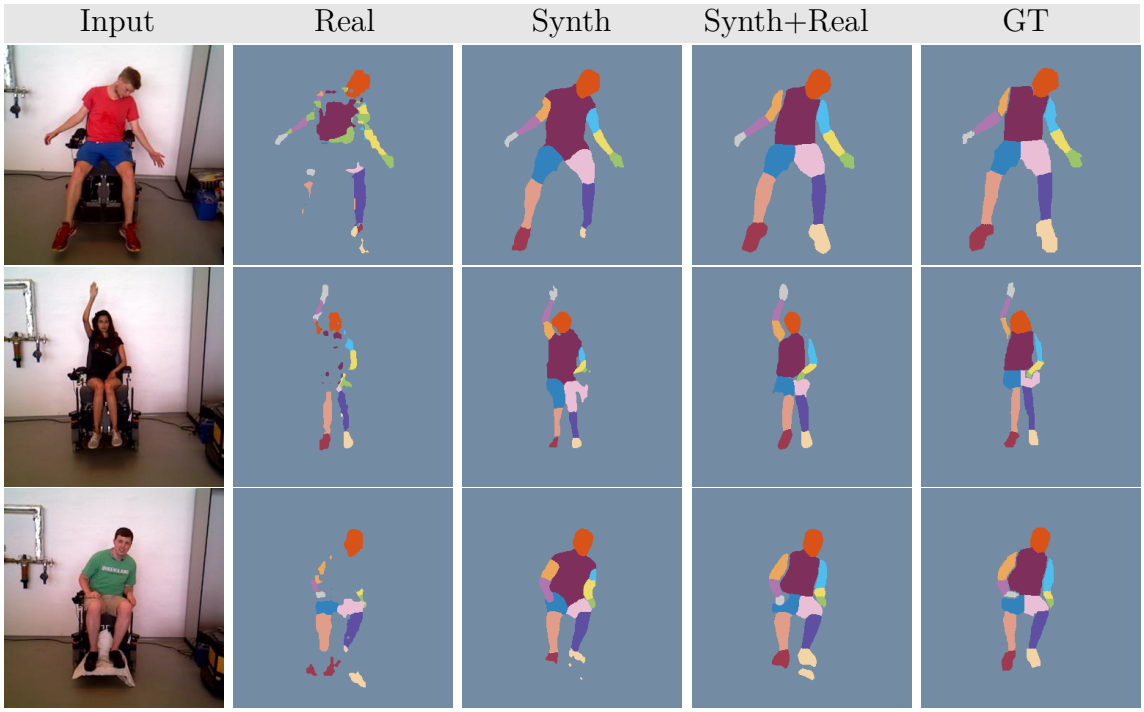


Figure 3-5 – Part segmentation on the Freiburg Sitting People dataset, training only on FSitting (Real), training only on synthetic images (Synth), fine-tuning on 2 training subjects from FSitting (Synth+Real). Fine-tuning helps although only for 200 iterations.

Table 3.2 – Parts segmentation results on 4 test subjects of Freiburg Sitting People dataset. IOU for head, torso and upper legs (averaged over left and right) are presented as well as the mean IOU and mean pixel accuracy over 14 parts. The means do not include background class. By adding an upsampling layer, we get the best results reported on this dataset.

Training data	Head IOU	Torso IOU	Legs_{up} IOU	mean IOU	mean Acc.
Real+Pascal [Oliveira et al., 2016]	-	-	-	64.10	81.78
Real	58.44	24.92	30.15	28.77	38.02
Synth	73.20	65.55	39.41	40.10	51.88
Synth+Real	72.88	80.76	65.41	59.58	78.14
Synth+Real+up	**85.09**	**87.91**	**77.00**	**68.84**	**83.37**

and st-RMSE errors, respectively. Figure 3-4 shows sample predictions. For both tasks, the results are mostly accurate on synthetic test images. However, there exist a few challenging poses (e.g. crawling), test samples with extreme close-up views, and fine details of the hands that are causing errors. In the following sections, we investigate if similar conclusions can be made for real images.

3.5.3 Segmentation on Freiburg Sitting People

Freiburg Sitting People (FSitting) dataset [Oliveira et al., 2016] is composed of 200 high resolution (300x300 pixels) front view images of 6 subjects sitting on a wheel chair. There are 14 human part annotations available. See Figure 3-5 for sample test images and corresponding ground truth (GT) annotation. We use the same train/test split as [Oliveira et al., 2016], 2 subjects for training and 4 subjects for test. The amount of data is limited for training deep networks. We show that our network pre-trained only on synthetic images is already able to segment human body parts. This shows that the human renderings in the synthetic dataset are representative of the real images, such that networks trained exclusively on synthetic data can generalize quite well to real data.

Table 3.2 summarizes segmentation results on FSitting. We carry out several ex-

Table 3.3 – Parts segmentation results on Human3.6M. The best result is obtained by fine-tuning synthetic network with real images. Although the performance of the network trained only with real data outperforms training only with synthetic, the predictions visually are worse because of overfitting, see Figure 3-6.

	IOU		Accuracy	
Training data	fg+bg	fg	fg+bg	fg
Real	49.61	46.32	58.54	55.69
Synth	46.35	42.91	56.51	53.55
Synth+Real	57.07	54.30	67.72	65.53

periments to understand the gain from synthetic pre-training. For the 'Real' baseline, we train a network from scratch using 2 training subjects. This network overfits as there are few subjects to learn from and the performance is quite low. Our 'Synth' result is obtained using the network pre-trained on synthetic images without fine-tuning. We get 51.88% pixel accuracy and 40.1% IOU with this method and clearly outperform training from real images. Furthermore, fine-tuning (Synth+Real) with 2 training subjects helps significantly. See Figure 3-5 for qualitative results. Given the little amount for training in FSitting, the fine-tuning converges after 200 iterations.

In [Oliveira et al., 2016], the authors introduce a network that outputs a high-resolution segmentation after several layers of upconvolutions. For a fair comparison, we modify our network to output full resolution by adding one bilinear upsampling layer followed by nonlinearity (ReLU) and a convolutional layer with 3×3 filters that outputs $15 \times 300 \times 300$ instead of $15 \times 64 \times 64$ as explained in Section 3.4. If we fine-tune this network (Synth+Real+up) on FSitting, we improve performance and outperform [Oliveira et al., 2016] by a large margin. Note that [Oliveira et al., 2016] trains on the same FSitting training images, but added around 2,800 Pascal images. Hence they use significantly more manual annotation than our method.

3.5.4 Segmentation and depth on Human3.6M

To evaluate our approach, we need sufficient real data with ground truth annotations. Such data is expensive to obtain and currently not available. For this reason, we generate nearly perfect ground truth for images recorded with a calibrated camera and given their MoCap data. Human3.6M [Ionescu et al., 2011, 2014b] is currently the largest dataset where such information is available. There are 3.6 million frames from 4 cameras. We use subjects S1, S5, S6, S7, S8 for training, S9 for validation and S11 for testing as in [Rogez and Schmid, 2016; Yasin et al., 2016], but from all 4 cameras. Note that this is different from the official train/test split [Ionescu et al., 2014b]. Each subject performs each of the 15 actions twice. We use all frames from one of the two instances of each action for training, and every 64^{th} frame from all instances for testing. The frames have resolution 1000×1000 pixels, we assume a 256×256 cropped human bounding box is given to reduce computational complexity. We evaluate the performance of both segmentation and depth, and compare with the baseline for which we train a network on real images only.

Segmentation. Table 3.3 summarizes the parts segmentation results on Human3.6M. Note that these are not comparable to the results in [Ionescu et al., 2014a] both because they assume the background segment is given and our ground truth segmentation data is not part of the official release (see Section 3.3.2). We report both the mean over 14 human parts (fg) and the mean together with the background class (fg+bg). Training on real images instead of synthetic images increases IOU by 3.4% and pixel accuracy by 2.14%. This is expected because the training distribution matches the test distribution in terms of background, camera position and action categories (i.e. poses). Furthermore, the amount of real data is sufficient to perform CNN training. However, since there are very few subjects available, we see that the network doesn't generalize to different clothing. In Figure 3-6, the 'Real' baseline has the border between shoulders and upper arms exactly on the T-shirt boundaries. This reveals that

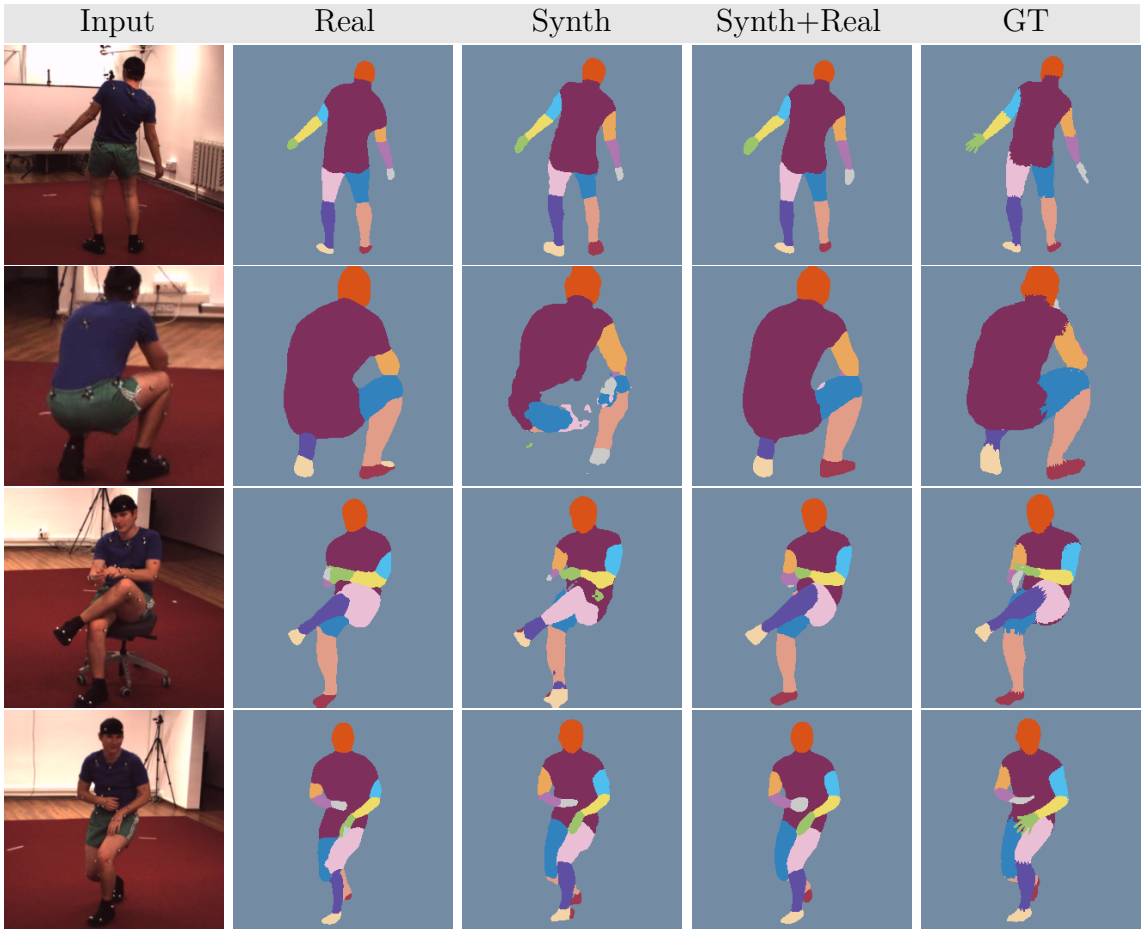


Figure 3-6 – Parts segmentation on the Human3.6M dataset, training only on real images and MoSH-generated ground-truth from Human3.6M (Real), training only on synthetic images from SURREAL (Synth), and fine-tuning on real Human3.6M data (Synth+Real). The 'Real' baseline clearly fails on upper arms by fitting the skin color. The synthetic pre-trained network has seen more variety in clothing. Best result is achieved by the fine-tuned network.

Figure 3-7 – Depth segmentation on the Human3.6M dataset, columns represent same training partitions as in Figure 3-6. The pre-trained network (Synth) fails due to scale mismatching in the training set and low contrast body parts, but fine-tuning with real data (Synth+Real) tends to recover from these problems.

Table 3.4 – Depth estimation results on Human3.6M (in millimeters). The depth errors RMSE and st-RMSE are reported on foreground pixels. PoseRMSE error is measured only on given human joints.

Training data	RMSE	st-RMSE	PoseRMSE	st-PoseRMSE
Real	96.3	75.2	122.6	94.5
Synth	111.6	98.1	152.5	131.5
Synth+Real	90.0	67.1	92.9	82.8

the network learns about skin color rather than actual body parts. Our pre-trained network (Synth) performs reasonably well, even though the pose distribution in our MoCap is quite different than that of Human3.6M. When we fine-tune the network with real images from Human3.6M (Synth+Real), the model predicts very accurate segmentations and outperforms the 'Real' baseline by a large margin. Moreover, our model is capable of distinguishing left and right most of the time on all 4 views since it has been trained with randomly sampled views.

Depth estimation. Depth estimation results on Human3.6M for various poses and viewpoints are illustrated in Figure 3-7. Here, the pre-trained network fails at the very challenging poses, although it still captures partly correct estimates (second row). Fine-tuning on real data compensates for these errors and refines estimations. In Table 3.4, we show RMSE error measured on foreground pixels, together with the scale-translation invariant version (see Section 3.5.1). We also report the error only on known 2D joints (PoseRMSE) to have an idea of how well a 3D pose estimation model would work based on the depth predictions. One would need to handle occluded joints to infer 3D locations of all joints, and this is beyond the scope of the current work.

3.5.5 Qualitative results on MPII Human Pose

FSitting and Human3.6M are relatively simple datasets with limited background clutter, few subjects, single person per image, full body visible. In this section, we test

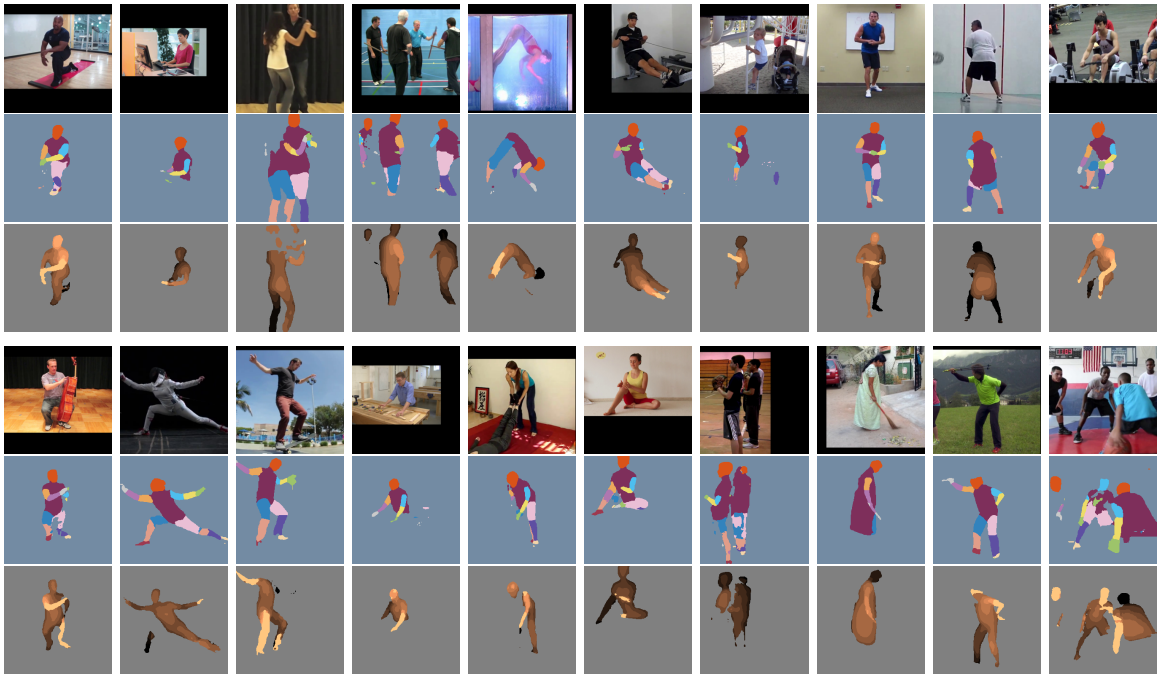


Figure 3-8 – Qualitative results on challenging images from MPII Human Pose dataset. Multi-person, occlusion and extreme poses are difficult cases for our model. Given that the model is trained only on synthetic data, it is able to generalize sufficiently well on cluttered real data. It is interesting to note that although we do not model cloth shape, we see in the 8th column (bottom) that the whole dress is labeled as torso and depth is quite accurate. Also the lower body occlusion never happens in training, but is handled well at test (2nd top, 4th bottom).

the generalization of our model on more challenging images. MPII Human Pose [Andriluka et al., 2014] is one of the largest datasets with diverse viewpoints and clutter. However, this dataset has no ground truth for part segmentation nor depth. Therefore, we qualitatively show our predictions. Figure 3-8 illustrates several success and failure cases. Our model generalizes reasonably well, except when there are multiple people close to each other and extreme viewpoints, which have not appeared during training. It is interesting to note that although lower body occlusions and cloth shapes are not present in synthetic training, the models perform accurately in such cases, see Figure 3-8 caption.

3.5.6 Design choices

We did several experiments to answer questions such as 'How much data should we synthesize?', 'Is CMU MoCap enough?', 'What's the effect having clothing variation?'.

Amount of data. We plot the performance as a function of training data size. We train with a random subset of 10^{-2}, 10^{-1}, 10^{0}, 10^{1}% of the 55K training clips using all frames of the selected clips, i.e., 10^{0}% corresponds to 550 clips with a total of 55k frames. Figure 3-9 (left) shows the increase in performance for both segmentation and depth as we increase training data. Results are plotted on synthetic and Human3.6M test sets with and without fine-tuning. The performance gain is higher at the beginning of all curves. There is some saturation, training with 55k frames is sufficient, and it is more evident on Human3.6M after a certain point. We explain this by the lack of diversity in Human3.6M test set and the redundancy of MoCap poses.

Clothing variation. Similarly, we study what happens when we add more clothing. We train with a subset of 100 clips containing only 1, 10 or 100 different clothings (out of a total of 930), because the dataset has maximum 100 clips for a given clothing

and we want to use same number of training clips, i.e., 1 clothing with 100 clips, 10 clothings with 10 clips each and 100 clothings with 1 clip each. Figure 3-9 (right) shows the increase in performance for both tasks as we increase clothing variation. In the case of fine-tuning, the impact gets less prominent because training and test images of Human3.6M are recorded in the same room. Moreover, there is only one subject in our test set, ideally such experiment should be evaluated on more diverse data.

MoCap variation. Pose distribution depends on the MoCap source. To experiment with the effect of having similar poses in training as in test, we rendered synthetic data using Human3.6M MoCap. Segmentation and depth networks pre-trained on this data (IOU: 48.11%, RMSE: 2.44) outperform the ones pre-trained on CMU MoCap (42.82%, 2.57) when tested on real Human3.6M. It is important to have diverse MoCap and to match the target distribution. Note that we exclude the Human3.6M synthetic data in Section 3.5.4 to address the more generic case where there is no dataset specific MoCap data available.

3.6 Conclusions

In this chapter, we have shown successful large-scale training of CNNs from synthetically generated images of people. We have addressed two tasks, namely, human body part segmentation and depth estimation, for which large-scale manual annotation is impractical. Unlike many existing synthetic datasets, the focus of SURREAL is on the realistic rendering of people, which is a challenging task. Our synthetic dataset has rich pixel-wise ground truth and can potentially be used for other tasks than considered here, e.g., see the next chapter on body shape estimation.

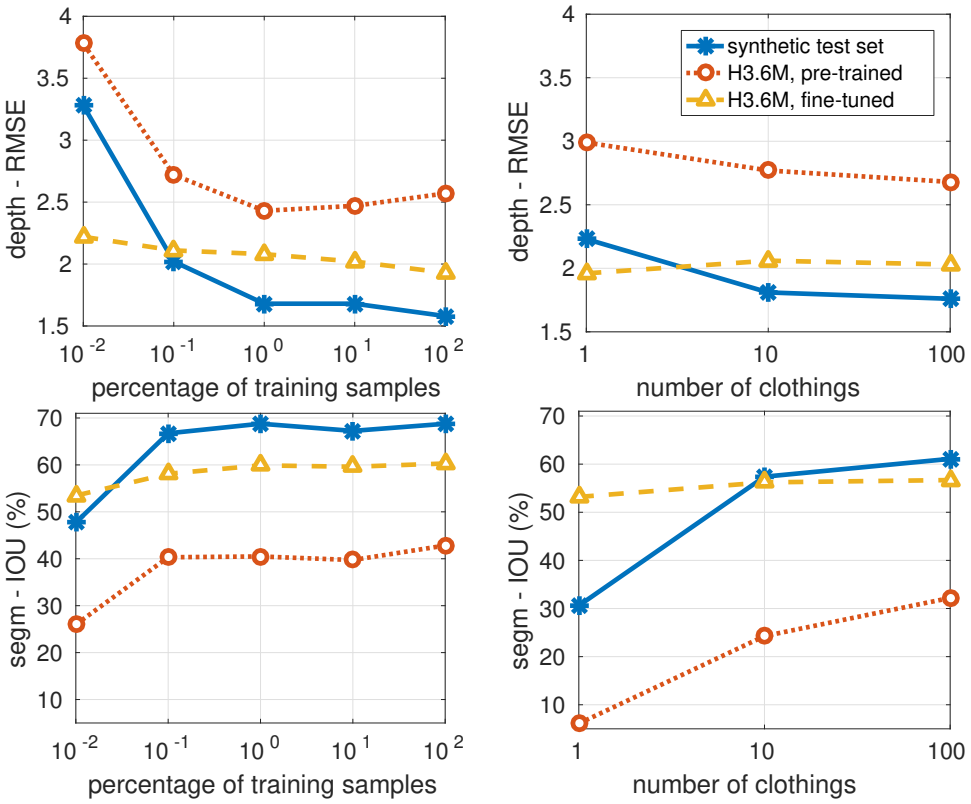


Figure 3-9 – **Left**: Amount of data. **Right**: Clothing variation. Segmentation and depth are tested on the synthetic and Human3.6M test sets with networks pre-trained on a subset of the synthetic training data. We also show fine-tuning on Human3.6M. The x-axis is in log-scale.

Chapter 4

BodyNet: Volumetric Inference of 3D Human Body Shapes

This chapter presents our second contribution on human body analysis. We address the 3D human body shape estimation problem and detail our approach.

Human shape estimation is an important task for video editing, animation and fashion industry. Predicting 3D human body shape from natural images, however, is highly challenging due to factors such as variation in human bodies, clothing and viewpoint. Prior methods addressing this problem typically attempt to fit parametric body models with certain priors on pose and shape. In this chapter, we argue for an alternative representation and propose BodyNet, a neural network for direct inference of volumetric body shape from a single image. BodyNet is an end-to-end trainable network that benefits from (i) a volumetric 3D loss, (ii) a multi-view re-projection loss, and (iii) intermediate supervision of 2D pose, 2D body part segmentation, and 3D pose. Each of them results in performance improvement as demonstrated by our experiments. To evaluate the method, we fit the SMPL model to our network output and show state-of-the-art results on the SURREAL and Unite the People datasets, outperforming recent approaches. Besides achieving state-of-the-art performance, our method also enables volumetric body part segmentation.

Figure 4-1 – Our BodyNet predicts a volumetric 3D human body shape and 3D body parts from a single image. We show the input image, the predicted human voxels, and the predicted part voxels.

4.1 Introduction

Parsing people in visual data is central to many applications including mixed-reality interfaces, animation, video editing and human action recognition. Towards this goal, human 2D pose estimation has been significantly advanced by recent efforts [Newell et al., 2016; Wei et al., 2016; Pishchulin et al., 2016; Cao et al., 2017]. Such methods aim to recover 2D locations of body joints and provide a simplified geometric representation of the human body. There has also been significant progress in 3D human pose estimation [Martinez et al., 2017; Pavlakos et al., 2017; Rogez et al., 2017; Zhou et al., 2017]. Many applications, however, such as virtual clothes try-on, video editing and re-enactment require accurate estimation of both 3D human *pose* and *shape*.

3D human shape estimation has been mostly studied in controlled settings using specific sensors including multi-view capture [Leroy et al., 2017], motion capture markers [Loper et al., 2014], inertial sensors [von Marcard et al., 2017], and 3D scanners [Yang et al., 2016]. In uncontrolled single-view settings 3D human shape estimation, however, has received little attention so far. The challenges include the lack of large-scale training data, the high dimensionality of the output space, and the choice of suitable representations for 3D human shape. Bogo et al. [2016] present the first automatic method to fit a deformable body model to an image but rely on accurate 2D pose estimation and introduce hand-designed constraints enforcing elbows and

knees to bend naturally. Other recent methods [Tan et al., 2017; Tung et al., 2017; Kanazawa et al., 2018a] employ deformable human body models such as SMPL [Loper et al., 2015] and regress model parameters with CNNs [Krizhevsky et al., 2012; LeCun et al., 1989]. In this work, we compare to such approaches and show advantages.

The optimal choice of 3D representation for neural networks remains an open problem. Recent work explores voxel [Maturana and Scherer, 2015; Yan et al., 2016; Yumer and Mitra, 2016; Girdhar et al., 2016], octree [Tatarchenko et al., 2017; Riegler et al., 2017b; Wang et al., 2017; Riegler et al., 2017a], point cloud [Su et al., 2017a,b; Deng et al., 2018], and surface [Groueix et al., 2018a] representations for modeling generic 3D objects. In the case of human bodies, the common approach has been to regress parameters of pre-defined human shape models [Tan et al., 2017; Tung et al., 2017; Kanazawa et al., 2018a]. However, the mapping between the 3D shape and parameters of deformable body models is highly nonlinear and is currently difficult to learn. Moreover, regression to a single set of parameters cannot represent multiple hypotheses and can be problematic in ambigous situations. Notably, skeleton regression methods for 2D human pose estimation, e.g., [Toshev and Szegedy, 2014], have recently been overtaken by heatmap based methods [Newell et al., 2016; Wei et al., 2016] enabling representation of multiple hypotheses.

In this work we propose and investigate a volumetric representation for body shape estimation as illustrated in Figure 4-1. Our network, called BodyNet, generates likelihoods on the 3D occupancy grid of a person. To efficiently train our network, we propose to regularize BodyNet with a set of auxiliary losses. Besides the main volumetric 3D loss, BodyNet includes a multi-view re-projection loss and multi-task losses. The multi-view re-projection loss, being efficiently approximated on voxel space (see Section 4.3.2), increases the importance of the boundary voxels. The multi-task losses are based on the additional intermediate network supervision in terms of 2D pose, 2D body part segmentation, and 3D pose. The overall architecture of BodyNet is illustrated in Figure 4-2.

To evaluate our method, we fit the SMPL model [Bogo et al., 2016] to the BodyNet output and measure single-view 3D human shape estimation performance in the recent SURREAL [Varol et al., 2017] (see Chapter 3) and Unite the People [Lassner et al., 2017] datasets. The proposed BodyNet approach demonstrates state-of-the-art performance and improves accuracy of recent methods. We show significant improvements provided by the end-to-end training and auxiliary losses of BodyNet. Furthermore, our method enables volumetric body-part segmentation. BodyNet is fully-differentiable and could be used as a subnetwork in future application-oriented methods targeting e.g., virtual cloth change or re-enactment.

In summary, this work makes several contributions. First, we address single-view 3D human shape estimation and propose a volumetric representation for this task. Second, we investigate several network architectures and propose an end-to-end trainable network BodyNet combining a multi-view re-projection loss together with intermediate network supervision in terms of 2D pose, 2D body part segmentation, and 3D pose. Third, we outperform previous regression-based methods and demonstrate state-of-the art performance on two datasets for human shape estimation. In addition, our network is fully differentiable and can provide volumetric body-part segmentation.

4.2 Related work

3D human body shape. While the problem of localizing 3D body joints has been well-explored in the past [Ionescu et al., 2014b; Kostrikov and Gall, 2014; Martinez et al., 2017; Pavlakos et al., 2017; Rogez et al., 2017; Yasin et al., 2016; Zhou et al., 2017; Rogez and Schmid, 2016], 3D human *shape* estimation from a single image has received limited attention and remains a challenging problem. For an overview on recent body shape estimation methods, see our literature review in Chapter 2, Section 2.1.4. Briefly, earlier work [Balan et al., 2007; Guan et al., 2009; Bogo et al., 2016;

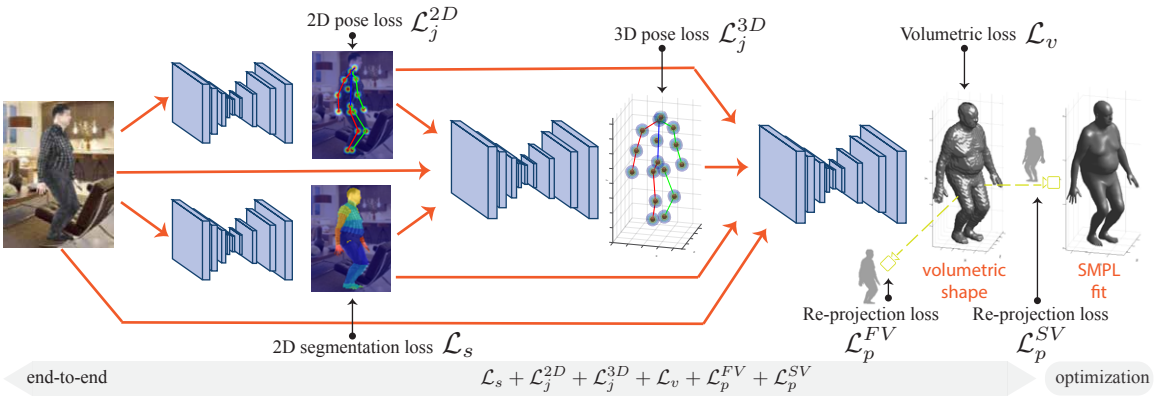


Figure 4-2 – BodyNet: End-to-end trainable network for 3D human body shape estimation. The input RGB image is first passed through subnetworks for 2D pose estimation and 2D body part segmentation. These predictions, combined with the RGB features, are fed to another network predicting 3D pose. All subnetworks are combined to a final network to infer volumetric shape. The 2D pose, 2D segmentation and 3D pose networks are first pre-trained and then fine-tuned jointly for the task of volumetric shape estimation using multi-view re-projection losses. We fit the SMPL model to volumetric predictions for the purpose of evaluation.

Lassner et al., 2017] focused on optimization-based methods to find the parameters of a deformable body model such as SCAPE [Anguelov et al., 2005] and SMPL [Loper et al., 2015]. Even though such methods show compelling results, inherently they are limited by the quality of the 2D detections they use and depend on priors both on pose and shape parameters to regularize the highly complex and costly optimization process. More recently, deep neural networks have been used to directly regress the model parameters [Tan et al., 2017; Tung et al., 2017; Kanazawa et al., 2018a].

Even though parameters of deformable body models provide a low-dimensional embedding of the 3D shape, predicting such parameters with a network requires learning a highly non-linear mapping. In our work we opt for an alternative volumetric representation that has shown to be effective for generic 3D objects [Yan et al., 2016] and faces [Jackson et al., 2017]. The approach of [Yan et al., 2016] operates on low-resolution grayscale images for a few rigid object categories such as chairs and tables. We argue that human bodies are more challenging due to significant non-

rigid deformations. To accommodate for such deformation, we use segmentation and 3D pose as proxy to 3D shape in addition to 2D pose [Jackson et al., 2017]. Conditioning our 3D shape estimation on a given 3D pose, the network focuses on the more complicated problem of shape deformation. Furthermore, we regularize our voxel predictions with additional re-projection loss, perform end-to-end multi-task training with intermediate supervision and obtain volumetric body part segmentation.

Others have studied predicting 2.5D projections of human bodies. DenseReg [Güler et al., 2017] and DensePose [Güler et al., 2018] estimate image-to-surface correspondences, while [Varol et al., 2017] outputs quantized depth maps for SMPL bodies. Differently from these methods, our approach generates a full 3D body reconstruction.

Multi-task neural networks. Multi-task networks are well-studied. A common approach is to output multiple related tasks at the very end of the neural network architecture. Another, more recently explored alternative is to stack multiple subnetworks and provide guidance with *intermediate supervision*. Here, we only cover related works that employ the latter approach. Guiding CNNs with relevant cues has shown improvements for a number of tasks. For example, 2D facial landmarks have shown useful guidance for 3D face reconstruction [Jackson et al., 2017] and similarly optical flow for action recognition [Simonyan and Zisserman, 2014]. However, these methods do not perform joint training. Recent work of [Luvizon et al., 2018] jointly learns 2D/3D pose together with action recognition. Similarly, [Popa et al., 2017] trains for 3D pose with intermediate tasks of 2D pose and segmentation. With this motivation, we make use of 2D pose, 2D human body part segmentation, and 3D pose, that provide cues for 3D human *shape* estimation. Unlike [Popa et al., 2017], 3D pose becomes an auxiliary task for our final 3D shape task. In our experiments, we show that training with a joint loss on all these tasks increases the performance of all our subnetworks (see Section 4.4.8).

4.3 BodyNet

BodyNet predicts 3D human body shape from a single image and is composed of four subnetworks trained first independently, then jointly to predict 2D pose, 2D body part segmentation, 3D pose, and 3D shape (see Figure 4-2). Here, we first discuss the details of the volumetric representation for body shape (Section 4.3.1). Then, we describe the multi-view re-projection loss (Section 4.3.2) and the multi-task training with the intermediate representations (Section 4.3.3). Finally, we formulate our model fitting procedure (Section 4.3.4).

4.3.1 Volumetric inference for 3D human shape

For 3D human body shape, we propose to use a voxel-based representation. Our shape estimation subnetwork outputs the 3D shape represented as an occupancy map defined on a fixed resolution voxel grid. Specifically, given a 3D body, we define a 3D voxel grid roughly centered at the root joint, (i.e., the hip joint) where each voxel inside the body is marked as occupied. We voxelize the ground truth meshes (i.e., SMPL) into a fixed resolution grid using binvox [Nooruddin and Turk, 2003; Min]. We assume orthographic projection and rescale the volume such that the xy-plane is aligned with the 2D segmentation mask to ensure spatial correspondence with the input image. After scaling, the body is centered on the z-axis and the remaining areas are padded with zeros.

Our network minimizes the binary cross-entropy loss after applying the sigmoid function on the network output similar to [Jackson et al., 2017]:

$$\mathcal{L}_v = \sum_{x=1}^{W}\sum_{y=1}^{H}\sum_{z=1}^{D} V_{xyz}\log\hat{V}_{xyz} + (1-V_{xyz})\log(1-\hat{V}_{xyz}), \tag{4.1}$$

where V_{xyz} and $\hat{V}_{xyz}$ denote the ground truth value and the predicted sigmoid output for a voxel, respectively. Width (W), height (H) and depth (D) are 128 in our

experiments. We observe that this resolution captures sufficient details.

The loss $\mathcal{L}_v$ is used to perform foreground-background segmentation of the voxel grid. We further extend this formulation to perform 3D body part segmentation with a multi-class cross-entropy loss. We define 6 parts (head, torso, left/right leg, left/right arm) and learn 7-class classification including the background. The weights for this network are initialized by the shape network by copying the output layer weights for each class. This simple extension allows the network to directly infer 3D body parts without going through the costly SMPL model fitting.

4.3.2 Multi-view re-projection loss on the silhouette

Due to the complex articulation of the human body, one major challenge in inferring the volumetric body shape is to ensure high confidence predictions across the whole body. We often observe that the confidences on the limbs away from the body center tend to be lower (see Figure 4-5). To address this problem, we employ additional 2D re-projection losses that increase the importance of the boundary voxels. Similar losses have been employed for rigid objects by [Zhu et al., 2017; Tulsiani et al., 2017] in the absence of 3D labels and by [Yan et al., 2016] as additional regularization. In our case, we show that the multi-view re-projection term is critical, particularly to obtain good quality reconstruction of body limbs. Assuming orthographic projection, the front view projection, $\hat{S}^{FV}$, is obtained by projecting the volumetric grid to the image with the *max* operator along the z-axis [Zhu et al., 2017]. Similarly, we define $\hat{S}^{SV}$ as the *max* along the x-axis:

$$\hat{S}^{FV}(x,y) = \max_z \hat{V}_{xyz} \quad \text{and} \quad \hat{S}^{SV}(y,z) = \max_x \hat{V}_{xyz}. \tag{4.2}$$

The true silhouette, S^{FV}, is defined by the ground truth 2D body part segmentation provided by the datasets. We obtain the ground truth side view silhouette from the voxel representation that we computed from the ground truth 3D mesh: $S^{SV}(y,z) =$

$\max_x V_{xyz}$. We note that our voxels remain slightly larger than the original mesh due to the voxelization step that marks every voxel that intersects with a face as occupied. We define a binary cross-entropy loss per view as follows:

$$\mathcal{L}_p^{FV} = \sum_{x=1}^{W}\sum_{y=1}^{H} S(x,y)\log \hat{S}^{FV}(x,y) + (1-S(x,y))\log(1-\hat{S}^{FV}(x,y)), \tag{4.3}$$

$$\mathcal{L}_p^{SV} = \sum_{y=1}^{H}\sum_{z=1}^{D} S(y,z)\log \hat{S}^{SV}(y,z) + (1-S(y,z))\log(1-\hat{S}^{SV}(y,z)). \tag{4.4}$$

We train the shape estimation network initially with $\mathcal{L}_v$. Then, we continue training with a combined loss: $\lambda_v\mathcal{L}_v + \lambda_p^{FV}\mathcal{L}_p^{FV} + \lambda_p^{SV}\mathcal{L}_p^{SV}$, Section 4.3.3 gives details on how to set the relative weighting of the losses. Section 4.4.3 demonstrates experimentally the benefits of the multi-view re-projection loss.

4.3.3 Multi-task learning with intermediate supervision

The input to the 3D shape estimation subnetwork is composed by combining RGB, 2D pose, segmentation, and 3D pose predictions. Here, we present the subnetworks used to predict these intermediate representations and detail our multi-task learning procedure. The architecture for each subnetwork is based on a stacked hourglass network [Newell et al., 2016], where the output is over a spatial grid and is, thus, convenient for pixel- and voxel-level tasks as in our case.

2D pose. Following the work of Newell et al. [2016], we use a heatmap representation of 2D pose. We predict one heatmap for each body joint where a Gaussian with fixed variance is centered at the corresponding image location of the joint. The final joint locations are identified as the pixel indices with the maximum value over each output channel. We use the first two stacks of an hourglass network to map RGB features $3\times256\times256$ to 2D joint heatmaps $16\times64\times64$ as in [Newell et al., 2016] and predict 16 body joints. The mean-squared error between the ground truth and predicted 2D heatmaps is $\mathcal{L}_j^{2D}$.

2D part segmentation. Our body part segmentation network is adopted from [Varol

et al., 2017] and is trained on the SMPL [Loper et al., 2015] anatomic parts defined by [Varol et al., 2017]. The architecture is similar to the 2D pose network and again the first two stacks are used. The network predicts one heatmap per body part given the input RGB image, which results in an output resolution of $15 \times 64 \times 64$ for 15 body parts. The spatial cross-entropy loss is denoted with $\mathcal{L}_s$.

3D pose. Estimating the 3D joint locations from a single image is an inherently ambiguous problem. To alleviate some uncertainty, we assume that the camera intrinsics are known and predict the 3D pose in the camera coordinate system. Extending the notion of 2D heatmaps to 3D, we represent 3D joint locations with 3D Gaussians defined on a voxel grid as in [Pavlakos et al., 2017]. For each joint, the network predicts a fixed-resolution volume with a single 3D Gaussian centered at the joint location. The $xy-$dimensions of this grid are aligned with the image coordinates, and hence the 2D joint locations, while the z dimension represents the depth. We assume this voxel grid is aligned with the 3D body such that the root joint corresponds to the center of the 3D volume. We determine a reasonable depth range in which a human body can fit (roughly 85cm in our experiments) and quantize this range into 19 bins. We define the overall resolution of the 3D grid to be $64 \times 64 \times 19$, i.e., four times smaller in spatial resolution compared to the input image as is the case for the 2D pose and segmentation networks. We define one such grid per body joint and regress with mean-squared error $\mathcal{L}_j^{3D}$.

The 3D pose estimation network consists of another two stacks. Unlike 2D pose and segmentation, the 3D pose network takes multiple modalities as input, all spatially aligned with the output of the network. Specifically, we concatenate RGB channels with the heatmaps corresponding to 2D joints and body parts. We upsample the heatmaps to match the RGB resolution, thus the input resolution becomes $(3 + 16 + 15) \times 256 \times 256$. While 2D pose provides a significant cue for the x, y joint locations, some of the depth information is implicitly contained in body part segmentation since unlike a silhouette, occlusion relations among individual body parts

provide strong 3D cues. For example a discontinuity on the torso segment caused by an occluding arm segment implies the arm is in front of the torso. In Section 4.4.8, we provide comparisons of 3D pose prediction with and without using this additional information.

Combined loss and training details. The subnetworks are initially trained independently with individual losses, then fine-tuned jointly with a combined loss:

$$\mathcal{L} = \lambda_j^{2D}\mathcal{L}_j^{2D} + \lambda_s\mathcal{L}_s + \lambda_j^{3D}\mathcal{L}_j^{3D} + \lambda_v\mathcal{L}_v + \lambda_p^{FV}\mathcal{L}_p^{FV} + \lambda_p^{SV}\mathcal{L}_p^{SV}. \tag{4.5}$$

The weighting coefficients are set such that the average gradient of each loss across parameters is at the same scale at the beginning of fine-tuning. With this rule, we set $(\lambda_j^{2D}, \lambda_s, \lambda_j^{3D}, \lambda_v, \lambda_p^{FV}, \lambda_p^{SV}) \propto (10^7, 10^3, 10^6, 10^1, 1, 1)$ and make the sum of the weights equal to one. We set these weights on the SURREAL dataset and use the same values in all experiments. We found it important to apply this balancing so that the network does not forget the intermediate tasks, but improves the performance of all tasks at the same time.

When training our full network, see Figure 4-2, we proceed as follows: (i) we train 2D pose and segmentation; (ii) we train 3D pose with fixed 2D pose and segmentation network weights; (iii) we train 3D shape network with all the preceding network weights fixed; (iv) then, we continue training the shape network with additional reprojection losses; (v) finally, we perform end-to-end fine-tuning on all network weights with the combined loss.

Implementation details. Each of our subnetworks consists of two stacks to keep a reasonable computational cost. We take the first two stacks of the 2D pose network trained on the MPII dataset [Andriluka et al., 2014] with 8 stacks [Newell et al., 2016]. Similarly, the segmentation network is trained on the SURREAL dataset with 8 stacks [Varol et al., 2017] and the first two stacks are used. Since stacked hourglass networks involve intermediate supervision [Newell et al., 2016], we can use

only part of the network by sacrificing slight performance. The weights for 3D pose and 3D shape networks are randomly initialized and trained on SURREAL with two stacks. Architectural details are given in Section 4.6. SURREAL [Varol et al., 2017], being a large-scale dataset, provides pre-training for the UP dataset [Lassner et al., 2017] where the networks converge relatively faster. Therefore, we fine-tune the segmentation, 3D pose, and 3D shape networks on UP from those pre-trained on SURREAL. We use RMSprop [Tieleman and Hinton, 2012] algorithm with mini-batches of size 6 and a fixed learning rate of 10^{-3}. Color jittering augmentation is applied on the RGB data. For all the networks, we assume that the bounding box of the person is given, thus we crop the image to center the person. Code is made publicly available on the project page [BodyNet project page].

4.3.4 Fitting a parametric body model

While the volumetric output of BodyNet produces good quality results, for some applications, it is important to produce a 3D surface mesh, or even a parametric model that can be manipulated. Furthermore, we use the SMPL model for our evaluation. To this end, we process the network output in two steps: (i) we first extract the isosurface from the predicted occupancy map, (ii) next, we optimize for the parameters of a deformable body model, SMPL model in our experiments, that fits the isosurface as well as the predicted 3D joint locations.

Formally, we define the set of 3D vertices in the isosurface mesh that is extracted [Lewiner et al., 2003] from the network output to be $\mathbf{V}^n$. SMPL [Loper et al., 2015] is a statistical model where the location of each vertex is given by a set $\mathbf{V}^s(\theta, \beta)$ that is formulated as a function of the pose (θ) and shape (β) parameters [Loper et al., 2015]. Given $\mathbf{V}^n$, our goal is to find $\{\theta^\star, \beta^\star\}$ such that the weighted Chamfer distance, i.e., the distance among the closest point correspondences between

$\mathbf{V}^n$ and $\mathbf{V}^s(\theta,\beta)$ is minimized:

$$\{\theta^\star,\beta^\star\} = \underset{\{\theta,\beta\}}{\operatorname{argmin}} \sum_{\mathbf{p}^n\in\mathbf{V}^n} \min_{\mathbf{p}^s\in\mathbf{V}^s(\theta,\beta)} w^n\|\mathbf{p}^n-\mathbf{p}^s\|_2^2 + \sum_{\mathbf{p}^s\in\mathbf{V}^s(\theta,\beta)} \min_{\mathbf{p}^n\in\mathbf{V}^n} w^n\|\mathbf{p}^n-\mathbf{p}^s\|_2^2 + \lambda\sum_{i=1}^{J}\|\mathbf{j}_i^n-\mathbf{j}_i^s(\theta,\beta)\|_2^2. \quad (4.6)$$

We find it effective to weight the closest point distances by the confidence of the corresponding point in the isosurface which depends on the voxel predictions of our network. We denote the weight associated with the point p^n as w^n. We define an additional term to measure the distance between the predicted 3D joint locations, $\{\mathbf{j}_i^n\}_{i=1}^J$, where J denotes the number of joints, and the corresponding joint locations in the SMPL model, denoted by $\{\mathbf{j}_i^s(\theta,\beta)\}_{i=1}^J$. We weight the contribution of the joints' error by a constant λ (empirically set to 5 in our experiments) since J is very small (e.g., 16) compared to the number of vertices (e.g., 6890). In Section 4.4, we show the benefits of fitting to voxel predictions compared to our baseline of fitting to 2D and 3D joints, and to 2D segmentation, i.e., to the inputs of the shape network.

We optimize for Eq. (4.6) in an iterative manner where we update the correspondences at each iteration. We use Powell's dogleg method [Nocedal and Wright, 2006] and Chumpy [Chumpy] similar to [Bogo et al., 2016]. When reconstructing the isosurface, we first apply a thresholding (0.5 in our experiments) to the voxel predictions and apply the marching cubes algorithm [Lewiner et al., 2003]. We initialize the SMPL pose parameters to be aligned with our 3D pose predictions and set $\beta = \vec{0}$ (where $\vec{0}$ denotes a vector of zeros).

4.4 Experiments

This section presents the evaluation of BodyNet. We first describe evaluation datasets (Section 4.4.1) and other methods used for comparison in this work (Section 4.4.2). We then evaluate contributions of additional inputs (Section 4.4.3) and losses (Sec-

tion 4.4.4). Next, we report performance on the UP dataset (Section 4.4.5). Finally, we demonstrate results for 3D body part segmentation (Section 4.4.6).

4.4.1 Datasets and evaluation measures

SURREAL dataset [Varol et al., 2017] is a large-scale synthetic dataset for 3D human body shapes with ground truth labels for segmentation, 2D/3D pose, and SMPL body parameters. Given its scale and rich ground truth, we use SURREAL in this work for training and testing. Previous work demonstrating successful use of synthetic images of people for training visual models include [Barbosa et al., 2018; Ghezelghieh et al., 2016; Chen et al., 2016b]. Given the SMPL shape and pose parameters, we compute the ground truth 3D mesh. We use the standard train split [Varol et al., 2017]. For testing, we use the middle frame of the middle clip of each test sequence, which makes a total of 507 images. We observed that testing on the full test set of $12,528$ images yield similar results. To evaluate the quality of our shape predictions for difficult cases, we define two subsets with extreme body shapes, similar to what is done for example in optical flow [Butler et al., 2012]. We compute the surface distance between the average shape ($\beta = \vec{0}$) given the ground truth pose and the true shape. We take the 10^{th} (*s10*) and 20^{th} (*s20*) percentile of this distance distribution that represent the meshes with extreme body shapes.

Unite the People dataset (UP) [Lassner et al., 2017] is a recent collection of multiple datasets (e.g., MPII [Andriluka et al., 2014], LSP [Johnson and Everingham, 2010]) providing additional annotations for each image. The annotations include 2D pose with 91 keypoints, 31 body part segments, and 3D SMPL models. The ground truth is acquired in a semi-automatic way and is therefore imprecise. We evaluate our 3D body shape estimations on this dataset. We report errors on two different subsets of the test set where 2D segmentations as well as pseudo 3D ground truth are available. We use notation T1 for images from the LSP subset [Lassner et al., 2017], and T2 for images used by [Tan et al., 2017].

3D shape evaluation. We evaluate body shape estimation with different measures. Given the ground truth and our predicted volumetric representation, we measure the intersection over union directly on the voxel grid, i.e., voxel IOU. We further assess the quality of the projected silhouette to enable comparison with [Lassner et al., 2017; Tan et al., 2017; Kanazawa et al., 2018a]. We report the intersection over union (silhouette IOU), F1-score computed for foreground pixels, and global accuracy (ratio of correctly predicted foreground and background pixels). We evaluate the quality of the fitted SMPL model by measuring the average error in millimeters between the corresponding vertices in the fit and ground truth mesh (surface error). We also report the average error between the corresponding 91 landmarks defined for the UP dataset [Lassner et al., 2017]. We assume the depth of the root joint and the focal length to be known to transform the volumetric representation into a metric space.

4.4.2 Alternative methods

We demonstrate advantages of BodyNet by comparing it to alternative methods. BodyNet makes use of 2D/3D pose estimation and 2D segmentation. We define alternative methods in terms of the same components combined differently.

SMPLify++. Lassner et al. [2017] extended SMPLify [Bogo et al., 2016] with an additional term on 2D silhouette. Here, we extend it further to enable a fair comparison with BodyNet. We use the code from [Bogo et al., 2016] and implement a fitting objective with additional terms on 2D silhouette and 3D pose besides 2D pose. Given the 2D silhouette contour predicted by the network $\mathbf{S}^n$, our goal is to find $\{\theta^\star, \beta^\star\}$ such that the weighted distance among the closest point correspondences between $\mathbf{S}^n$ and $\mathbf{S}^s(\theta, \beta)$ is minimized:

$$\{\theta^\star, \beta^\star\} = \underset{\{\theta,\beta\}}{\operatorname{argmin}} \sum_{\mathbf{p}^s \in \mathbf{S}^s(\theta,\beta)} \min_{\mathbf{p}^n \in \mathbf{S}^n} w^n \|\mathbf{p}^n - \mathbf{p}^s\|_2^2 + \lambda_j \sum_{i=1}^{J} \|\mathbf{j}_i^n - \mathbf{j}_i^s(\theta,\beta)\|_2^2 + \sum_{i=1}^{J} \|\mathbf{k}_i^n - \mathbf{k}_i^s(\theta,\beta)\|_2^2, \tag{4.7}$$

where $\mathbf{S}^s(\theta, \beta)$ is the projected silhouette of the SMPL model. Prior to the optimization, we initialize the camera parameters with the original function from SMPLify [Bogo et al., 2016] that only uses the hip and shoulder joints for an estimate. We use this function for initialization and further optimize the camera parameters using our 2D/3D joint correspondences. We use these camera parameters to compute the projection. The weights w^n associated to the contour point p^n denote the pixel distance between p^n and its closest point (divided by the pixel threshold 10, defined by [Bogo et al., 2016]).

Similar to Eq. (4.6), the second term measures the distance between the predicted 3D joint locations, $\{\mathbf{j}_i^n\}_{i=1}^J$, where J denotes the number of joints, and the corresponding joint locations in the SMPL model, denoted by $\{\mathbf{j}_i^s(\theta, \beta)\}_{i=1}^J$. Additionally, we define predicted 2D joint locations, $\{\mathbf{k}_i^n\}_{i=1}^J$, and 2D SMPL joint locations, $\{\mathbf{k}_i^s(\theta, \beta)\}_{i=1}^J$. We set the weight $\lambda_j = 100$ by visual inspection. We observe that it becomes difficult to tune the weights with multiple objectives. We optimize for Eq. (4.7) in an iterative manner where we update the correspondences at each iteration.

As shown in Table 4.2, results of SMPLify++ remain inferior to BodyNet despite both of them using 2D/3D pose and segmentation inputs (see Figure 4-3).

Shape parameter regression. To validate our volumetric representation, we also implement a regression method by replacing the 3D shape estimation network in Figure 4-2 by another subnetwork directly regressing the 10-dim. shape parameter vector β using L2 loss. The network architecture corresponds to the encoder part of the hourglass followed by 3 additional fully connected layers (see Section 4.6 for details). We recover the pose parameters θ from our 3D pose prediction (initial attempts to regress θ together with β gave worse results). Table 4.2 demonstrates inferior performance of the β regression network that often produces average body shapes (see Figure 4-3). In contrast, BodyNet results in better SMPL fitting due to the accurate volumetric representation.

Table 4.1 – Performance on the SURREAL dataset using alternative combinations of intermediate representations at the input.

	voxel IOU (%)	SMPL surface error (mm)
2D pose	47.7	80.9
RGB	51.8	79.1
Segm	54.6	79.1
3D pose	56.3	74.5
Segm + 3D pose	56.4	74.0
RGB + 2D pose + Segm + 3D pose	**58.1**	**73.6**

4.4.3 Effect of additional inputs

We first motivate our proposed architecture by evaluating performance of 3D shape estimation in the SURREAL dataset using alternative inputs (see Table 4.1). When only using one input, 3D pose network, which is already trained with additional 2D pose and segmentation inputs, performs best. We observe improvements as more cues, specifically 3D cues are added. We also note that intermediate representations in terms of 3D pose and 2D segmentation outperform RGB. Adding RGB to the intermediate representations further improves shape results on SURREAL. Figure 4-4 illustrates intermediate predictions as well as the final 3D shape output. Based on

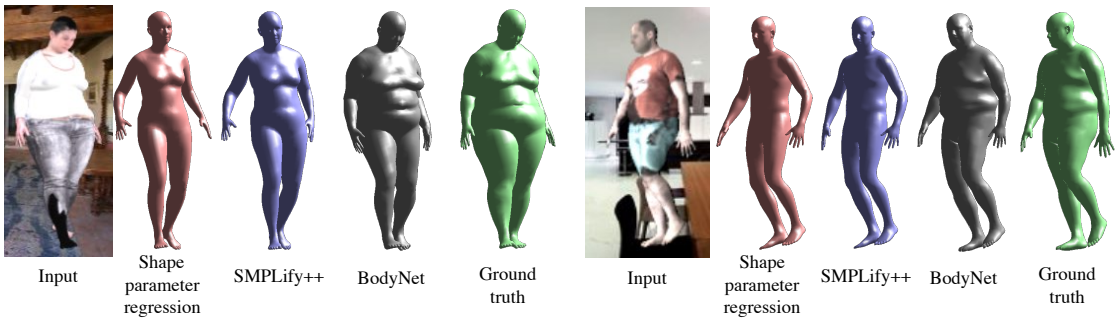


Figure 4-3 – SMPL fit on BodyNet predictions compared with other methods. While shape parameter regression and the fitting only to BodyNet inputs (SMPLify++) produce shapes close to average, BodyNet learns how the true shape observed in the image deviates from the average deformable shape model. Examples taken from the test subset *s10* of SURREAL dataset with extreme shapes.

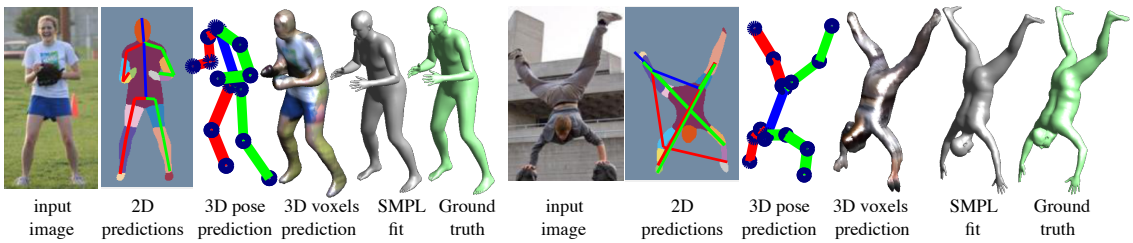


Figure 4-4 – Our predicted 2D pose, segmentation, 3D pose, 3D volumetric shape, and SMPL model alignments. Our 3D shape predictions are consistent with pose and segmentation, suggesting that the shape network relies on the intermediate representations. When one of the auxiliary tasks fails (2D pose on the right), 3D shape can still be recovered with the help of the other cues.

results in Table 4.1, we choose to use all intermediate representations as parts of our full network that we call BodyNet.

4.4.4 Effect of re-projection error and end-to-end multi-task training

We evaluate contributions provided by additional supervision from Section 4.3.2-4.3.3.

Effect of re-projection losses. Table 4.2 (lines 4-10) provides results when the shape network is trained with and without re-projection losses (see also Figure 4-5). The voxels network without any additional loss already outperforms the baselines described in Section 4.4.2. When trained with re-projection losses, we observe increasing performance both with single-view constraints, i.e., front view (FV), and multi-view, i.e., front and side views (FV+SV). The multi-view re-projection loss puts more importance on the body surface resulting in a better SMPL fit.

Effect of intermediate losses. Table 4.2 (lines 7-10) presents experimental evaluation of the proposed intermediate supervision. Here, we first compare the end-to-end network fine-tuned jointly with auxiliary tasks (lines 9-10) to the networks trained independently from the fixed representations (lines 4-6). Comparison of results on lines 6 and 10 suggests that multi-task training regularizes all subnetworks and provides better performance for 3D shape. We refer to Section 4.4.8 for the performance

Table 4.2 – Volumetric prediction on SURREAL with different versions of our model compared to alternative methods. Note that lines 2-10 use same modalities (i.e., 2D/3D pose, 2D segmentation). The evaluation is made on the SMPL model fit to our voxel outputs. The average SMPL surface error decreases with the addition of the proposed components.

		full	*s20*	*s10*
1.	Tung et al. [2017] (using GT 2D pose and segmentation)	74.5	-	-
	Alternative methods:			
2.	SMPLify++ (θ, β optimized)	75.3	79.7	86.1
3.	Shape parameter regression (β regressed, θ fixed)	74.3	82.1	88.7
	BodyNet:			
4.	Voxels network	73.6	81.1	86.3
5.	Voxels network with [FV] silhouette re-projection	69.9	76.3	81.3
6.	Voxels network with [FV+SV] silhouette re-projection	68.2	74.4	79.3
7.	End-to-end without intermediate tasks [FV]	72.7	78.9	83.2
8.	End-to-end without intermediate tasks [FV+SV]	70.5	76.9	81.3
9.	End-to-end with intermediate tasks [FV]	67.7	74.7	81.0
10.	End-to-end with intermediate tasks [FV+SV]	**65.8**	**72.2**	**76.6**

improvements on auxiliary tasks. To assess the contribution of intermediate losses on 2D pose, segmentation, and 3D pose, we implement an additional baseline where we again fine-tune end-to-end, but remove the losses on the intermediate tasks (lines 7-8). Here, we keep only the voxels and the re-projection losses. These networks not only forget the intermediate tasks, but are also outperformed by our base networks without end-to-end refinement (compare lines 8 and 6). On all the test subsets (i.e., full, *s20*, and *s10*) we observe a consistent improvement of the proposed components against baselines. Figure 4-3 presents qualitative results and illustrates how BodyNet successfully learns the 3D shape in extreme cases.

Comparison to the state of the art. Table 4.2 (lines 1,10) demonstrates a significant improvement of BodyNet compared to the recent method of Tung et al. [2017]. Note that [Tung et al., 2017] relies on ground truth 2D pose and segmentation on the test set, while our approach is fully automatic. Other works do not report results on

Table 4.3 – Body shape performance and comparison to the state of the art on the UP dataset. Unlike in SURREAL, the 3D ground truth in this dataset is imprecise. [1]This result is reported in [Lassner et al., 2017]. [2]This result is reported in [Tan et al., 2017].

		2D metrics			*3D metrics (mm)*	
		Acc. (%)	IOU	F1	Landmarks	Surface
T1	3D ground truth [Lassner et al., 2017]	92.17	-	0.88	0	0
	Decision forests [Lassner et al., 2017]	86.60	-	0.80	-	-
	HMR [Kanazawa et al., 2018a]	91.30	-	0.86	-	-
	SMPLify, UP-P91 [Lassner et al., 2017]	90.99	-	0.86	-	-
	SMPLify on DeepCut [Bogo et al., 2016][1]	91.89	-	**0.88**	-	-
	BodyNet *(end-to-end multi-task)*	92.75	**0.73**	0.84	**83.3**	**102.5**
T2	3D ground truth [Lassner et al., 2017][2]	95.00	0.82	-	0	0
	Indirect learning [Tan et al., 2017]	**95.00**	**0.83**	-	190.0	-
	Direct learning [Tan et al., 2017]	91.00	0.71	-	105.0	-
	BodyNet *(end-to-end multi-task)*	92.97	0.75	**0.86**	**69.6**	**80.1**

Table 4.4 – 2D metrics on the UP dataset to compare manual segmentations (M-network) versus SMPL projections (S-network) as re-projection supervision. [1]This result is reported in [Lassner et al., 2017].

		Acc. (%)	IOU	F1
T1	SMPLify on DeepCut [Bogo et al., 2016][1]	91.89	-	0.88
	S-network *(SMPL projections)*	92.75	0.73	0.84
	M-network *(manual segmentations)*	**94.67**	**0.80**	**0.89**
T2	Indirect learning [Tan et al., 2017]	95.00	**0.83**	-
	S-network *(SMPL projections)*	92.97	0.75	0.86
	M-network *(manual segmentations)*	**95.11**	0.82	**0.90**

the recent SURREAL dataset.

4.4.5 Comparison to the state of the art on Unite the People

For the networks trained on the UP dataset, we initialize the weights pre-trained on SURREAL and fine-tune with the complete training set of UP-3D where the 2D segmentations are obtained from the provided 3D SMPL fits [Lassner et al., 2017]. We show results of BodyNet trained end-to-end with multi-view re-projection loss. We provide quantitative evaluation of our method in Table 4.3 and compare to recent approaches [Tan et al., 2017; Kanazawa et al., 2018a; Lassner et al., 2017]. We

note that some works only report 2D metrics measuring how well the 3D shape is aligned with the manually annotated segmentation. The ground truth is a noisy estimate obtained in a semi-automatic way [Lassner et al., 2017], whose projection is mostly accurate but not its depth. While our results are on par with previous approaches on 2D metrics, we note that the provided manual segmentations and the 3D SMPL fits [Lassner et al., 2017] are noisy and affect both the training and the evaluation [Güler et al., 2018]. Therefore, we also provide a large set of visual results in Appendices 4.5, 4.4.7 to illustrate our competitive 3D estimation quality. On 3D metrics, our method significantly outperforms both direct and indirect learning of [Tan et al., 2017]. We also provide qualitative results in Figure 4-4 where we show both the intermediate outputs and the final 3D shape predicted by our method. We observe that voxel predictions are aligned with the 3D pose predictions and provide a robust SMPL fit. We refer to Section 4.4.7 for an analysis on the type of segmentation used as re-projection supervision.

4.4.6 3D body part segmentation

As described in Section 4.3.1, we extend our method to produce not only the foreground voxels for a human body, but also the 3D part labeling. We report quantitative results on SURREAL in Table 4.5 where accurate ground truth is available. When the parts are combined, the foreground IOU becomes 58.9 which is comparable to 58.1 reported in Table 4.1. We provide qualitative results in Figure 4-6 on the UP dataset where the parts network is only trained on SURREAL. To the best of our knowledge, we present the first method for 3D body part labeling from a single image with an end-to-end approach. We infer volumetric body parts directly with a network without iterative fitting of a deformable model and obtain successful results. Performance-wise BodyNet can produce foreground and per-limb voxels in 0.28s and 0.58s per image, respectively, using modern GPUs.

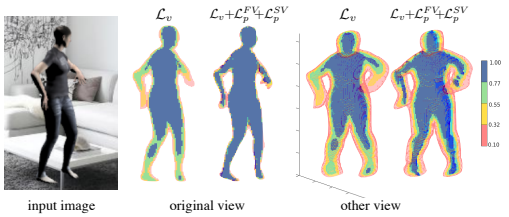


Figure 4-5 – Voxel predictions color-coded based on the confidence values. Notice that our combined 3D and re-projection loss enables our network to make more confident predictions across the whole body. Example taken from SURREAL.

Figure 4-6 – BodyNet is able to directly regress volumetric body parts from a single image on examples from UP.

Table 4.5 – 3D body part segmentation performance measured per part on SURREAL. The articulated and small limbs appear more difficult than torso.

	Head	Torso	Left arm	Right arm	Left leg	Right leg	Background	Foreground
Voxel IOU (%)	49.8	67.9	29.6	28.3	46.3	46.3	99.1	58.9

4.4.7 Potential to capture cloth deformations

Our experiments up to this point do not use the manual segmentation of the UP dataset for training, although the evaluation on 2D metrics is performed against this ground truth. Here we experiment with the option of using manual annotations for the front view re-projection loss (M-network) versus the SMPL projections (S-network) as supervision. Table 4.4 summarizes results. We obtain significantly better aligned silhouettes with M-network by using the manual annotations during training. However; in this case, the volumetric supervision is not in agreement with the 2D re-projection loss. We observe that this problem creates artifacts in the output 3D shape. Figure 4-7 illustrates this effect. We show results from both M-network and S-network. Note that while the cloth boundaries are better captured with the M-network from the front view, the output appears noisy from a rotated view.

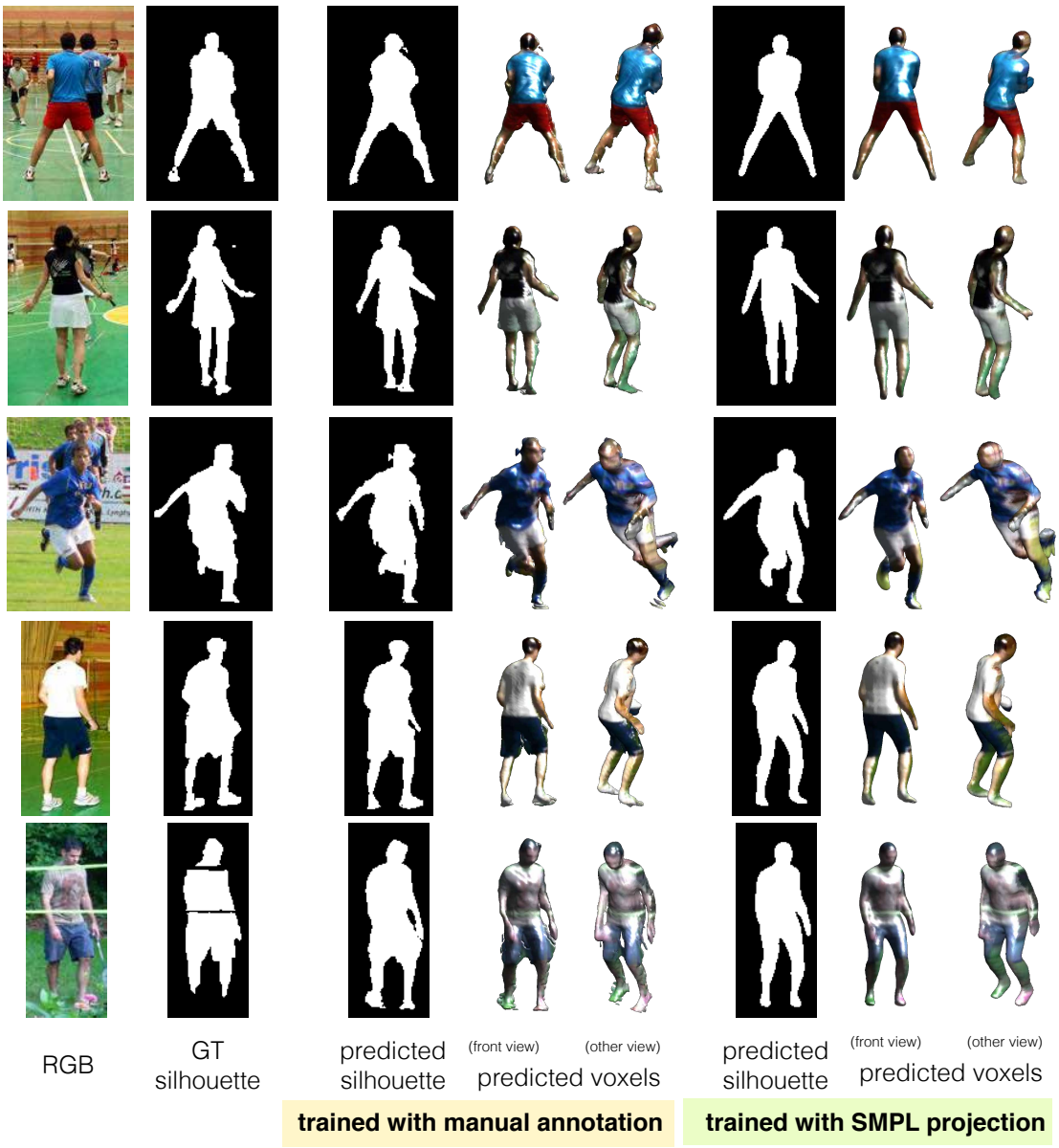


Figure 4-7 – Using manual segmentations (M-network) versus SMPL projections (S-network) as re-projection supervision on the UP dataset.

Table 4.6 – Performances of intermediate tasks before and after end-to-end multi-task fine-tuning on the SURREAL dataset. All 2D pose, segmentation and 3D pose results improve with the joint training.

	Segmentation mean parts IOU (%)	2D pose PCKh@0.5	3D pose mean joint distance (mm)
Independent single-task training	59.2	82.7	46.1
Joint multi-task training	**69.2**	**90.8**	**40.8**

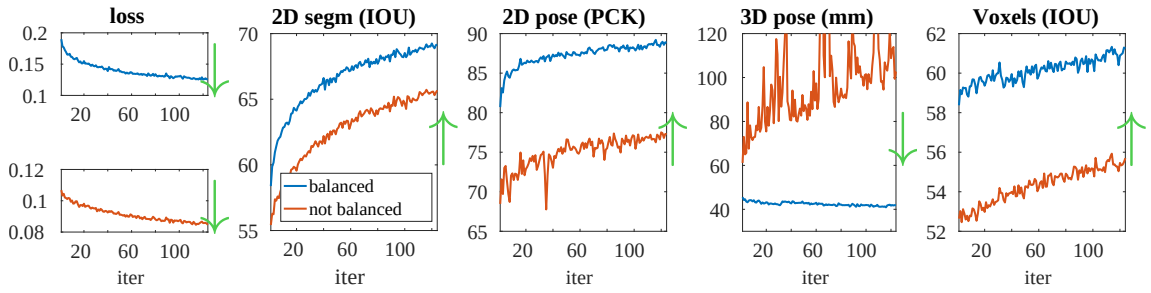


Figure 4-8 – Training curves with (blue) and without (red) multi-task loss balancing. Loss and the performance of each task are plotted at every 2K iterations. Balancing is crucial to equally learn all tasks, see especially 3D pose that is balanced with a factor 10^6.

4.4.8 Performance of intermediate tasks

Effect of multi-task training. Table 4.6 reports the results before and after end-to-end training for 2D pose, segmentation, and 3D pose (lines 6 and 10 of Table 4.2). We report mean IOU of the 14 foreground parts (excluding the background) as in [Varol et al., 2017] for segmentation performance. 2D pose performance is measured with PCKh@0.5 as in [Newell et al., 2016]. We measure the 3D pose error averaged over 16 joints in millimeters. We report the error of our predictions against ground truth with both of them centered at the root joint. We further assume the depth of the root joint to be given in order to convert xy components of our volumetric 3D pose representation in pixel space into metric space. The joint training for all tasks improves both the accuracy of 3D shape estimation as well as the performance of all intermediate tasks.

Table 4.7 – Performance of our segmentation subnetwork on the UP dataset. See text for details.

	avg macro F1
Trained with LSP SMPL projections [Lassner et al., 2017]	0.5628
Trained with the manual annotations [Lassner et al., 2017]	0.6046
Trained with full training (31 parts) [Lassner et al., 2017]	0.6101
Trained with full training (14 parts), pre-trained on SURREAL (ours)	**0.6397**

Balancing multi-task losses. We set the weights in the multi-task loss by bringing the gradients of individual losses to the same scale (see Section 4.3.3). For this, we set all weights to be equal (sum to 1) and run the training for 100 iterations. We then average the gradient magnitudes and find relative ratios to scale individual losses. In Figure 4-8, we show the training curves with and without such balancing.

2D segmentation subnetwork on the UP dataset. We give details on how the segmentation network that is pre-trained on SURREAL is fine-tuned on the UP dataset. Furthermore, we report the performance and compare it to [Lassner et al., 2017].

The segmentation network of BodyNet requires 15 classes (14 body parts and the background). On the UP dataset, there are several types of segmentation annotations. The training set of UP-3D has 5703 images with 31 part labels obtained from the projections of the automatically generated SMPL ground truth. Manual segmentation of six body parts only exists for the LSP subset of 639 images out of the full 1000 images with manual segmentations (not all have SMPL ground truth). We group the 31 SMPL parts into 14, which changes the definition of some part boundaries slightly, but are quickly learned during fine-tuning. With this strategy, we obtain 5703 training images. Figure 4-9 shows qualitative results for the segmentation capability of our network. For quantitative evaluation, we use the full 1000 LSP images and group our 14 parts into 6. We report macro F1 score that is averaged over 6 parts and the background as in [Lassner et al., 2017]. Table 4.7 compares to other results reported in [Lassner et al., 2017]. Our subnetwork demonstrates state-of-the-art results.

Figure 4-9 – Qualitative 2D body part segmentation results on the UP dataset.

Effect of additional inputs for 3D pose. In this section, we motivate the initial layers of the BodyNet architecture. Specifically, we investigate the effect of using different input combinations of RGB, 2D pose, and 2D segmentation for the 3D pose estimation task. For this experiment, we do not perform end-to-end fine-tuning (similar to Table 4.1). Table 4.8 shows the effect of gradually adding more cues at the input level and demonstrates consistent improvements on two different datasets. Here, we report results on both SURREAL and the widely used 3D pose benchmark of Human3.6M dataset [Ionescu et al., 2014b]. We fine-tune our networks which are pre-trained on SURREAL by using sequences from subjects S1, S5, S6, S7, S8, S9 and evaluate on every 64th frame of camera 2 recording subject S11 (i.e., *protocol 1* described in [Rogez et al., 2017]).

We compare our 3D pose estimation with the state-of-the-art methods in Table 4.8. Note that unlike others, we do not apply any rotation transformation on our output before evaluation and we assume the depth of the root joint to be known. While these are therefore not directly comparable, our approach achieves state-of-the-art performance on the Human3.6M dataset.

Table 4.8 – 3D pose error (mm) of our 3D pose prediction network when different intermediate representations are used as input. Notice that combining all input cues yields best results, which achieves state of the art.

Input	SURREAL	Human3.6M
RGB	49.1	51.6
2D pose	55.9	57.0
Segm	48.1	58.9
2D pose + Segm	47.7	56.3
RGB + 2D pose + Segm	**46.1**	**49.0**
Kostrikov and Gall [2014]		115.7
Yasin et al. [2016]		108.3
Rogez and Schmid [2016]		88.1
Rogez et al. [2017]		53.4

4.5 Qualitative analysis

Volumetric shape results. We illustrate additional examples of BodyNet output in Figure 4-15 and in the video available in the project page [BodyNet project page]. We show original RGB images with corresponding predictions of 3D volumetric body shapes. For the visualization we threshold the real-valued 3D output of the BodyNet using 0.5 as threshold and show the fitted surface [Lewiner et al., 2003]. The texture on reconstructed bodies is automatically segmented and mapped from original images. We also show additional examples of SMPL fits and 3D body part segmentations. For the part segmentation, each voxel is assigned to the part with the maximum score and an isosurface is shown per body part. Results are shown for static images from the Unite the People dataset [Lassner et al., 2017] and on a few real videos from YouTube. Notably, the method obtains temporally consistent results even when applied to individual frames of the video (see video in the project page [BodyNet project page] between 2:20-2:45).

Predicted silhouettes versus manual segmentations on UP. Figure 4-10 compares projected silhouettes of our voxel predictions (middle) with the manually an-

notated segmentations (right) used as ground truth for the evaluation in Table 4.3. While our results are as expected and good, we observe frequent inconsistencies with

Figure 4-10 – Projected silhouettes of our voxel predictions (middle) versus manually annotated segmentations (right) on the Unite the People dataset. We note that the evaluation is problematic due to several reasons such as occlusions and clothings, whose definitions are application-dependent, i.e., one might be interested in anatomic body or the full cloth deformation.

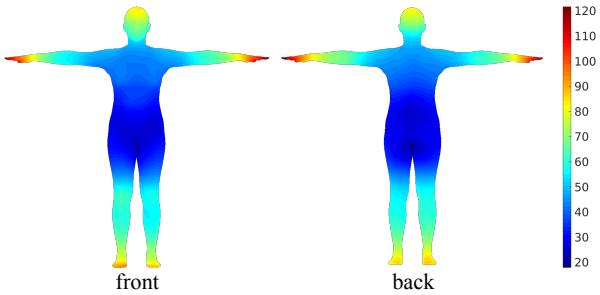


Figure 4-11 – Per-vertex SMPL error on SURREAL. Hands and feet contribute the most to the surface error, followed by the other articulated body parts.

the manual annotation due to several reasons: BodyNet produces a full 3D human body shape even in the case of occlusions (blue); annotations are often imprecise (red); the 3D prediction of cloth (green) and hair (yellow) is currently beyond this work due to the lack of training data, we instead focus on producing the anatomic parts (e.g., two legs instead of a long dress); finally, the labels are not always consistent in the case of multi-person images (purple).

We note that we never use manual segmentation during training as such annotations are not available for the full UP-3D dataset. As supervision for re-projection losses we instead use the SMPL silhouettes whose overlap with the manual segmentation is already not perfect (see Table 4.3, first row). Therefore, our performance in 2D metrics has an upper bound. Due to difficulties with the quantitative evaluation, we mostly rely on qualitative results for the UP dataset.

SMPL error. We next investigate the quality of predictions depending on the body location. We examine the network from Table 4.2 (line 10, 65.8mm surface error) and measure the average per-vertex error. We visualize the color-coded SMPL surface in Figure 4-11 indicating the areas with the highest and lowest errors by the red and blue colors, respectively. Unsurprisingly, the highest errors occur at the extremities of the body which can be explained by the articulation and the limited resolution of the voxel grid preventing the capture of fine details.

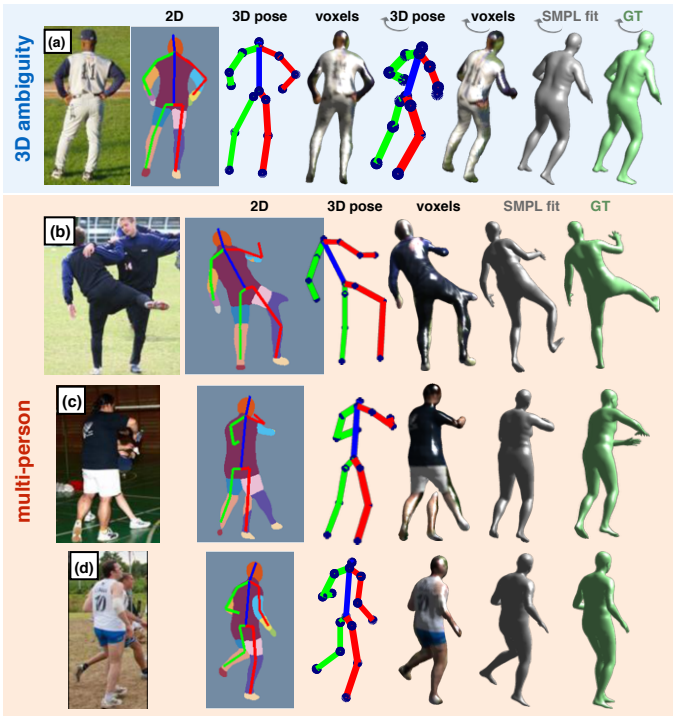


Figure 4-12 – Failure cases on images from UP. Arrows denote rotated views. Top(a): results for depth ambiguity visible with the rotated view. Bottom(b-d): intermediate predictions failing in a multi-person image. Note that GT is inaccurate due to the semi-automatic annotation protocol.

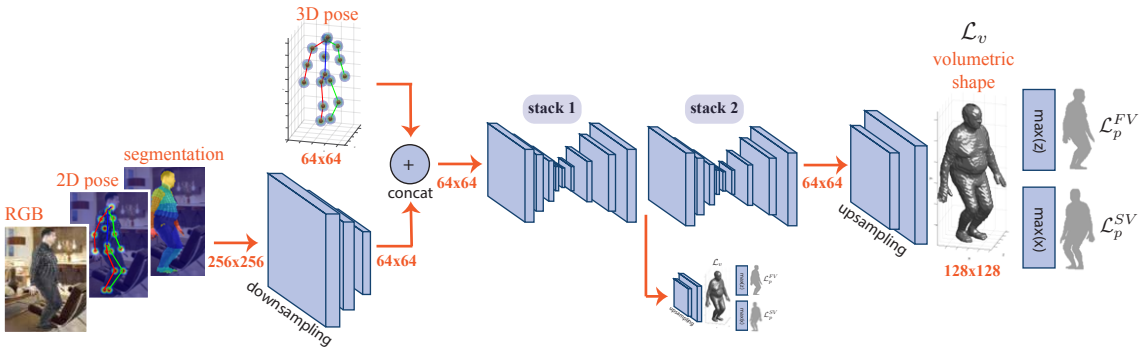


Figure 4-13 – Detailed architecture of our volumetric shape estimation subnetwork of BodyNet. The resolutions denoted in red refer to the *spatial* resolution. See text for details.

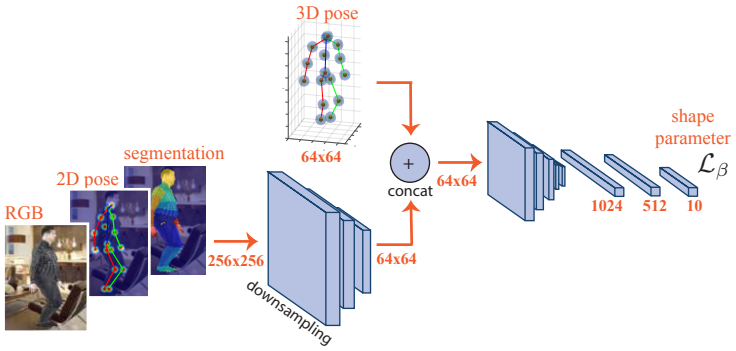


Figure 4-14 – Detailed architecture of the shape parameter regression subnetwork described in Section 4.4.2.

Failure modes. Figure 4-12 presents failure cases on UP. Depth ambiguity (a) and multi-person images (b-d) often cause failures of pose estimation that propagate further to the voxel output. Note that UP GT also has errors and our method may learn such errors when trained on UP.

4.6 Architecture details

Volumetric shape network. The architecture for our 3D shape estimation network is detailed in Figure 4-13. As described in Section 4.3.1, this network consists of

two hourglasses, each supervised by the same type of losses. Different than the other subnetworks in BodyNet, the input to the shape estimation network is a combination of multiple modalities of different resolutions. We design an architecture whose first branch operates on the concatenation of RGB ($3\times256\times256$), 2D pose ($16\times256\times256$), and segmentation ($15 \times 256 \times 256$) channels as done in the original stacked hourglass network [Newell et al., 2016] where a series of convolution and pooling operations downsample the spatial resolution with a factor of 4. Once the spatial resolution of this branch matches the one of the 3D pose input, i.e., 64×64, we concatenate the feature maps of the first branch with the 3D pose heatmap channels. Note that the depth resolution of 3D pose is treated as input channels, thus its dimensions become $304 \times 64 \times 64$ for 16 body joints and 19 depth units ($304 = 16 \times 19$). The output of the second hourglass has again 64×64 spatial resolution. We use bilinear upsampling followed by ReLU and 3×3 convolutions to obtain the output resolution of $128 \times 128 \times 128$.

Shape parameter regression network. We described shape parameter regression as an alternative method in Section 4.4.2. Figure 4-14 gives architectural details for this subnetwork. The input part of the network is the same as in Figure 4-13. The output resolution at the bottleneck layer of the hourglass is $128 \times 4 \times 4$ (i.e., 2048-dim). We vectorize this output and add 3 fully connected layers of size $fc1(2048, 1024)$, $fc2(1024, 512)$ and $fc3(512, 10)$ to produce the 10-dim β vector with shape parameters of the SMPL [Loper et al., 2015] body model. This subnetwork is trained with the L2 loss.

3D body part segmentation network. When extending our shape network to produce 3D body parts as described in Section 4.3.1, we first copy the weights of the shape network trained without any re-projection loss (line 4 of the Table 4.2). We first train this network for 3D body parts and then fine-tune it with the additional multi-view re-projection losses. We apply one re-projection loss per part and

per view, i.e., 7*2=14 binary cross-entropy losses for 6 parts and 1 background, for frontal and side views. For 6 parts, we apply the *max* operation as in Section 4.3.2. For the background class, we apply the *min* operation to approximate orthographic projection.

4.7 Conclusion

We have presented BodyNet, a fully automatic end-to-end multi-task network architecture that predicts the 3D human body shape from a single image. We have shown that joint training with intermediate tasks significantly improves the results. We have also demonstrated that the volumetric regression together with a multi-view re-projection loss is effective for representing human bodies. Moreover, with this flexible representation, our framework allows us to extend our approach to demonstrate impressive results on 3D body part segmentation from a single image. We believe that BodyNet can provide a trainable building block for future methods that make use of 3D body information, such as virtual cloth-change and action recognition. Furthermore, we believe exploring the limits of using only intermediate representations is an interesting research direction for 3D tasks where acquiring training data is impractical. Another future direction is to study the 3D body shape under clothing. Volumetric representation can potentially capture such additional geometry if training data is provided.

In the next part of the thesis, we concentrate on the human action recognition problem, which is closely related to human body analysis. Instead of the static body pose, we will consider dynamic videos and a higher-level understanding of action categories, such as running and jumping.

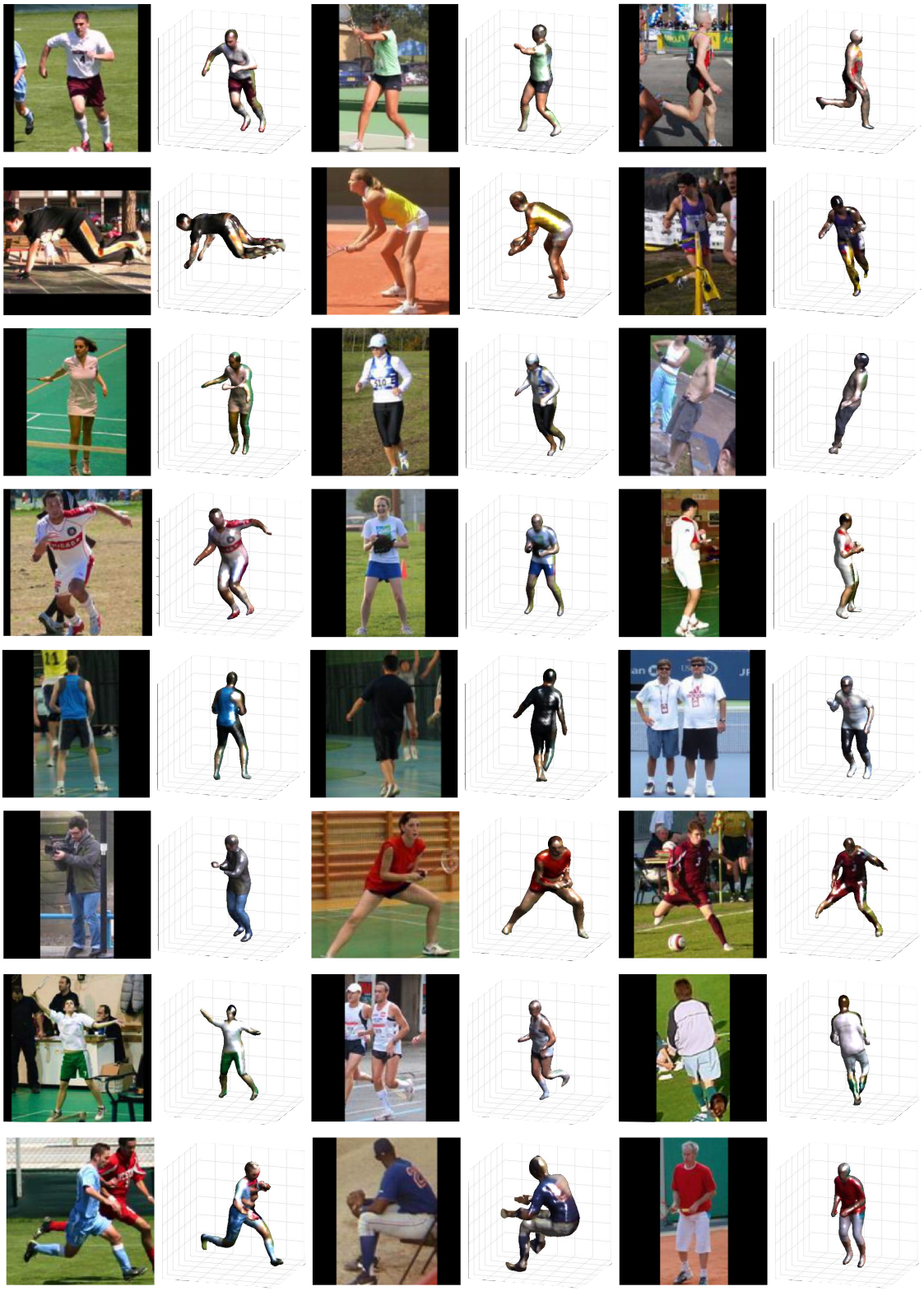


Figure 4-15 – Qualitative results of our volumetric shape predictions on UP.

Part II

Video Representations for Human Action Recognition

Chapter 5

Long-term Temporal Convolutions for Action Recognition

This chapter presents our first contribution to human action recognition. Here, we detail our approach to long-term temporal convolutions.

Typical human actions last several seconds and exhibit characteristic spatio-temporal structure. Recent methods attempt to capture this structure and learn action representations with convolutional neural networks. Such representations, however, are typically learned at the level of a few video frames failing to model actions at their full temporal extent. In this chapter, we learn video representations using neural networks with long-term temporal convolutions (LTC). We demonstrate that LTC-CNN models with increased temporal extents improve the accuracy of action recognition. We also study the impact of different low-level representations, such as raw values of video pixels and optical flow vector fields and demonstrate the importance of high-quality optical flow estimation for learning accurate action models. We report state-of-the-art results on two challenging benchmarks for human action recognition UCF101 (92.7%) and HMDB51 (67.2%).

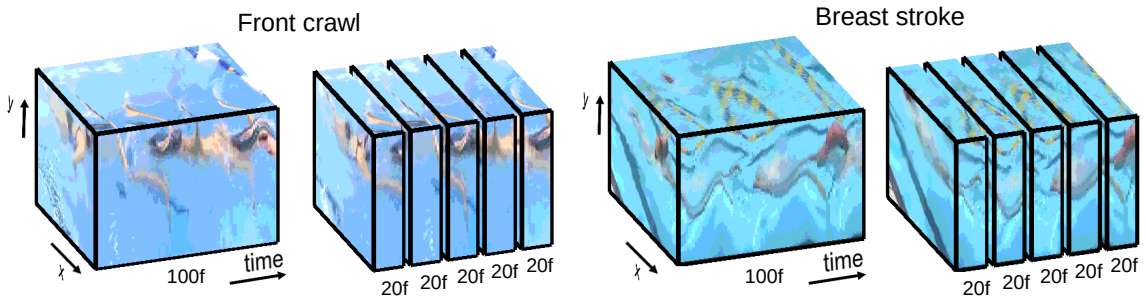


Figure 5-1 – Video patches for two classes of swimming actions. (a),(c): Actions often contain characteristic, class-specific space-time patterns that last for several seconds. (b),(d): Splitting videos into short temporal intervals is likely to destroy such patterns making recognition more difficult. Our neural network with Long-term Temporal Convolutions (LTC) learns video representations over extended periods of time.

5.1 Introduction

Human actions and events can be seen as spatio-temporal objects. Such a view finds support both in psychology [Tversky et al., 2002] and in computer vision approaches to action recognition in video [Laptev et al., 2008; Niebles et al., 2008; Schüldt et al., 2004; Wang and Schmid, 2013]. Successful methods for action recognition, indeed, share similar techniques with object recognition and represent actions by statistical models of local video descriptors. Differently to objects, however, actions are characterized by the temporal evolution of appearance governed by motion. Consistent with this fact, motion-based video descriptors such as HOF and MBH [Laptev et al., 2008; Wang and Schmid, 2013] as well as recent CNN-based motion representations [Simonyan and Zisserman, 2014] have shown most gains for action recognition in practice.

The recent rise of convolutional neural networks (CNNs) convincingly demonstrates the power of learning visual representations [Krizhevsky et al., 2012]. Equipped with large-scale training datasets [Deng et al., 2009; Zhou et al., 2014], CNNs have quickly taken over the majority of still-image recognition tasks such as object, scene and face recognition [Girshick et al., 2014; Taigman et al., 2014; Zhou et al., 2014]. Ex-

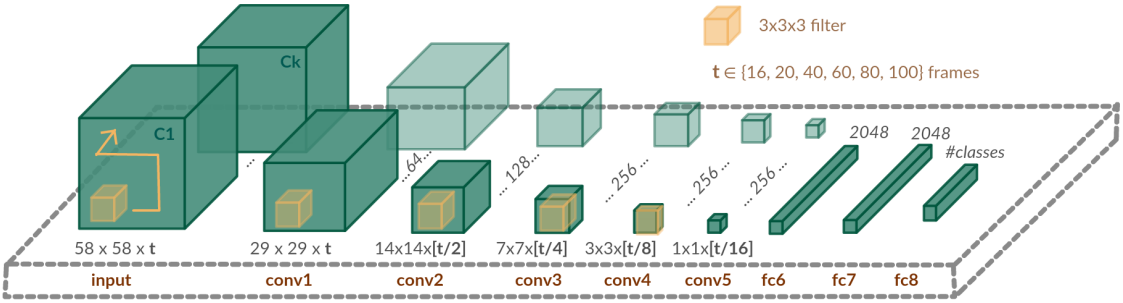


Figure 5-2 – Network architecture. Spatio-temporal convolutions with 3x3x3 filters are applied in the first 5 layers of the network. Max pooling and ReLU are applied in between all convolutional layers. Network input channels $C1...Ck$ are defined for different temporal resolutions $t \in \{20, 40, 60, 80, 100\}$ and either two-channel motion (*flow-x*, *flow-y*) or three-channel appearance (R,G,B). The spatio-temporal resolution of convolution layers decreases with the pooling operations.

tensions of CNNs to action recognition in video have been proposed in several recent works [Karpathy et al., 2014; Simonyan and Zisserman, 2014; Tran et al., 2015]. Such methods, however, currently show only moderate improvements over earlier methods using hand-crafted video features [Wang and Schmid, 2013].

Current CNN methods for action recognition often extend CNN architectures for static images [Krizhevsky et al., 2012] and learn action representations for short video intervals ranging from 1 to 16 frames [Karpathy et al., 2014; Simonyan and Zisserman, 2014; Tran et al., 2015]. Yet, typical human actions such as hand-shaking and drinking, as well as cycles of repetitive actions such as walking and swimming often last several seconds and span tens or hundreds of video frames. As illustrated in Figure 5-1(a),(c), actions often contain characteristic patterns with specific spatial as well as *long-term* temporal structure. Breaking this structure into short clips (see Figure 5-1(b),(d)) and aggregating video-level information by the simple average of clip scores [Simonyan and Zisserman, 2014; Tran et al., 2015] or more sophisticated schemes such as LSTMs [Donahue et al., 2015] is likely to be suboptimal.

In this work, we investigate the learning of long-term video representations. We consider space-time convolutional neural networks [Ji et al., 2010; Taylor et al., 2010;

Tran et al., 2015] and study architectures with Long-term Temporal Convolutions (LTC), see Figure 5-2. To keep the complexity of networks tractable, we increase the temporal extent of representations at the cost of decreased spatial resolution. We also study the impact of different low-level representations, such as raw values of video pixels and optical flow vector fields. Our experiments confirm the advantage of motion-based representations and highlight the importance of good quality motion estimation for learning efficient representations for human action recognition. We report state-of-the-art performance on two recent and challenging human action benchmarks: UCF101 and HMDB51.

The contributions of this work are twofold. We demonstrate *(i)* the advantages of long-term temporal convolutions and *(ii)* the importance of high-quality optical flow estimation for learning accurate video representations for human action recognition. In the remaining part of the chapter we discuss related work in Section 5.2, describe space-time CNN architectures in Section 5.3 and present an extensive experimental study of our method in Section 5.4. Our implementation and pre-trained CNN models (compatible with Torch) are available on the project web page [LTC project page].

5.2 Related Work

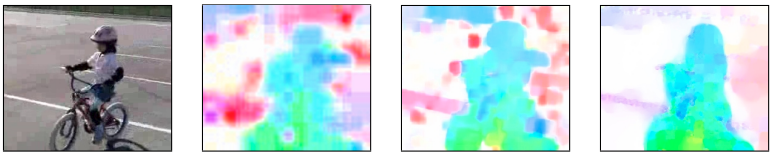
We review previous work on action recognition in our literature survey (Chapter 2, Section 2.2). Here, we refer to some of the most relevant CNN-based action recognition methods at the time of this work.

Learning visual representations with CNNs [Krizhevsky et al., 2012; LeCun et al., 1989] has shown clear advantages over "hand-crafted" features for many recognition tasks in static images [Girshick et al., 2014; Taigman et al., 2014; Zhou et al., 2014]. Extensions of CNN representations to action recognition in video have been proposed in several recent works [Donahue et al., 2015; Ji et al., 2010; Karpathy et al., 2014; Simonyan and Zisserman, 2014; Taylor et al., 2010; Tran et al., 2015; Wang et al.,

2015a,b; Bilen et al., 2016; Feichtenhofer et al., 2016]. Some of these methods encode single video frames with static CNN features [Donahue et al., 2015; Karpathy et al., 2014; Simonyan and Zisserman, 2014]. Extensions to short video clips where video frames are treated as multi-channel inputs to 2D CNNs have also been investigated in [Karpathy et al., 2014; Simonyan and Zisserman, 2014; Feichtenhofer et al., 2016; Wang et al., 2015b].

Learning CNN representations for action recognition has been addressed for raw pixel inputs and for pre-computed optical flow features. Consistent with previous results obtained with hand-crafted representations, motion-based CNNs typically outperform CNN representations learned for RGB inputs [Simonyan and Zisserman, 2014; Wang et al., 2015b]. In this work we investigate multi-resolution representations of motion and appearance where for motion-based CNNs we demonstrate the importance of high-quality optical flow estimation. Similar findings have been recently confirmed by [Zhang et al., 2016], where the authors transfer knowledge from high quality optical flow algorithms to motion vector encoding representation.

Most of the current CNN methods use architectures with 2D convolutions, enabling shift-invariant representations in the image plane. Meanwhile, the invariance to translations in time is also important for action recognition since the beginning and the end of actions is unknown in general. CNNs with 3D spatio-temporal convolutions address this issue and provide a natural extension of 2D CNNs to video. 3D CNNs have been investigated for action recognition in [Ji et al., 2010; Karpathy et al., 2014; Taylor et al., 2010; Tran et al., 2015]. All of these methods, however, learn video representations for RGB input. Moreover, they typically consider very short video intervals, for example, 16-frame video clips are used in [Tran et al., 2015] and 2, 7, 15 frames in [Taylor et al., 2010; Ji et al., 2010; Karpathy et al., 2014], respectively. In this work we extend 3D CNNs to significantly longer temporal convolutions that enable action representation at their full temporal scale. We also explore the impact of optical flow input. Both of these extensions show clear advantages in



RGB MPEG flow Farneback Brox

Input	Clip	Video
RGB	57.0	59.9
MPEG flow	58.5	63.8
Farneback	66.3	71.3
Brox	**74.8**	**79.6**

Figure 5-3 – Illustration of the three optical flow methods and comparison of corresponding recognition performance. From left to right: original image, MPEG [Kantorov and Laptev, 2014], Farneback [Farnebäck, 2003] and Brox [Brox et al., 2004] optical flow. The color coding indicates the orientation of the flow. The table on the right presents accuracy of action recognition in UCF101 (split 1) for different inputs. Results are obtained with 60f networks and training from scratch (see text for more details).

our experimental comparison to previous methods.

5.3 Long-term Temporal Convolutions

In this section we first present the network architecture. We then specify the different inputs to networks used in this work. We finally provide details on learning and testing procedures.

5.3.1 Network architecture

Our network architecture with long-term temporal convolutions is illustrated in Figure 5-2. The network has 5 space-time convolutional layers with 64, 128, 256, 256 and 256 filter response maps, followed by 3 fully connected layers of sizes 2048, 2048 and the number of classes. Following [Tran et al., 2015] we use $3 \times 3 \times 3$ space-time filters for all convolutional layers. Each convolutional layer is followed by a rectified linear unit (ReLU) and a space-time max pooling layer. Max pooling filters are of size $2 \times 2 \times 2$ except in the first layer, where it is $2 \times 2 \times 1$. The size of convolution output is kept constant by padding 1 pixel in all three dimensions. Filter stride for all dimensions is 1 for convolution and 2 for pooling operations. We use dropout for the first two fully connected layers. Fully connected layers are followed by ReLU layers.

Softmax layer at the end of the network outputs class scores.

5.3.2 Network input

To investigate the impact of long-term temporal convolutions, we here study network inputs with different temporal extents. We depart from the recent C3D work [Tran et al., 2015] and first compare inputs of 16 frames (16f) and 60 frames (60f). We then systematically analyze implications of the increased temporal and spatial resolutions for input signals in terms of motion and appearance. For the 16-frame network we crop input patches of size $112 \times 112 \times 16$ from videos with spatial resolution 171×128 pixels. We choose this baseline architecture to enable direct comparison with [Tran et al., 2015]. For the 60-frames networks we decrease spatial resolution to preserve network complexity and use input patches of size $58 \times 58 \times 60$ randomly cropped from videos rescaled to 89×67 spatial resolution.

As illustrated in Figure 5-2, the temporal resolution in our 60f network corresponds to 60, 30, 15, 7 and 3 frames for each of the five convolutional layers. In comparison, the temporal resolution of the 16f network is reduced more drastically to 16, 8, 4, 2 and 1 frame at each convolutional layer. We believe that preserving the temporal resolution at higher convolutional layers should enable learning more complex temporal patterns. The space-time resolution for the outputs of the fifth convolutional layers is $3 \times 3 \times 1$ and $1 \times 1 \times 3$ for the 16f and 60f networks respectively. The two networks have a similar number of parameters in the *fc6* layer and the same number of parameters in all other layers. For a systematic study of networks with different input resolutions we also evaluate the effect of increased temporal resolution $t \in \{20, 40, 60, 80, 100\}$ and varying spatial resolution of $\{58 \times 58, 71 \times 71\}$ pixels.

In addition to the input size, we experiment with different types of input modalities. First, as in [Tran et al., 2015], we use raw RGB values from video frames as input. To explicitly learn motion representations, we also use flow fields in x and y directions as input to our networks. Flow is computed for original videos. To

maintain correct flow values for network inputs with reduced spatial resolution, the magnitude of the flow is scaled by the factor of spatial subsampling. In other words, if a point moves 2 pixels in a 320×240 video frame, its motion will be 1 pixel when the frame is resized to 160×120 resolution. Moreover, to center the input data, we follow [Simonyan and Zisserman, 2014] and subtract the mean flow vector for each frame.

To investigate the dependency of action recognition on the quality of motion estimation, we experiment with three types of flow inputs obtained either directly from the video encoding, referred to as MPEG flow [Kantorov and Laptev, 2014], or from two optical flow estimators, namely Farneback [Farnebäck, 2003] and Brox [Brox et al., 2004]. Figure 5-3 shows results for the three flow algorithms. MPEG flow is a fast substitute for optical flow which we obtain from the original video encoding. Such flow, however, has low spatial resolution. It also misses flow vectors at some frames (I-frames) which we interpolate from neighboring frames. Farneback flow is also relatively fast and obtains rather noisy flow estimates. The approach of Brox flow is the most sophisticated of the three and is known to perform well in various flow estimation benchmarks.

5.3.3 Learning

We train our networks on the training set of each split independently for both UCF101 and HMDB51 datasets, which contain 9.5K and 3.7K videos, respectively. We use stochastic gradient descent applied to mini-batches with negative log likelihood criterion. For 16f networks we use a mini-batch size of 30 video clips. We reduce the batch size to 15 video clips for 60f networks, and 10 clips for 100f networks due to limitations of our GPUs. The initial learning rate for networks learned from scratch is 3×10^{-3} and 3×10^{-4} for networks fine-tuned from pre-trained models. For UCF101, the learning rate is decreased twice with a factor of 10^{-1}. For 16f networks, the first decrease is after 80K iterations and the second one after 45K additional iterations.

The optimization is completed after 20K more iterations. Convergence is faster for HMDB51, so the learning rate is decreased once after 60K iterations and completed after 10K more iterations. These numbers are doubled for 60f networks and tripled for 100f networks, since their batch sizes are twice and three times smaller compared to 16f nets. The above schedule is used together with 0.9 dropout ratio. Our experimental setups with 0.5 dropout ratio have less iterations due to faster convergence. The momentum is set to 0.9 and weight decay is initialized with 5×10^{-3} and reduced by a factor of 10^{-1} at every decrease of the learning rate.

Inspired by the random spatial cropping during training, we apply the corresponding augmentation to the temporal dimension as in [Simonyan and Zisserman, 2014], which we call *random clipping*. During training, given an input video, we randomly select a point (x, y, t) to sample a video clip of fixed size. A common alternative is to preprocess the data by using a sliding window approach to have pre-segmented clips of fixed size; however, this approach limits the amount of data when the windows are not overlapped as in [Tran et al., 2015]. Another data augmentation method that we evaluate is to have a multiscale cropping similar to [Wang et al., 2015b]. For this, we randomly select a coefficient for width and height separately from (1.0, 0.875, 0.75, 0.66) and resize the cropped region to the size of the network input. Finally, we horizontally flip the input with 50% probability.

At test time, a video is divided into t-frame clips with a temporal stride of 4 frames. Each clip is further tested with 10 crops, namely the 4 corners and the center, together with their horizontal flips. The video score is obtained by averaging over clip scores and crop scores. If the number of frames in a video is less than the clip size, we pad the input by repeating the last frames to fill the missing volume.

5.4 Experiments

We perform experiments on two widely used and challenging benchmarks for action recognition: UCF101 and HMDB51 (Section 5.4.1). We first examine the effect of network parameters (Section 5.4.2). We then compare to the state of the art (Section 5.4.3) and present a visual analysis of the spatio-temporal filters (Section 5.4.4). Finally we report runtime analysis (Section 5.4.5).

5.4.1 Datasets and evaluation metrics

UCF101 Soomro et al. [2012] is a widely-used benchmark for action recognition with 13K clips from YouTube videos lasting 7 seconds on average. The total number of frames is 2.4M distributed among 101 categories. The videos have spatial resolution of 320×240 pixels and 25 fps frame rate.

The HMDB51 dataset Kuehne et al. [2011] consists of 7K videos of 51 actions. The videos have 320×240 pixels spatial resolution and 30 fps frame rate. Although this dataset has been considered a large-scale benchmark for action recognition for the past few years, the amount of data for learning deep networks is limited.

We rely on two evaluation metrics. The first one measures per-clip accuracy, i.e. we assign each clip the class label with the maximum softmax output and measure the number of correctly assigned labels over all clips. The second metric measures video accuracy, i.e. the standard evaluation protocol. To obtain a video score we average the per-clip softmax scores and take the maximum value of this average as class label. We average over all videos to obtain video accuracy. We report our final results according to the standard evaluation protocol, which is the mean video accuracy across the three test splits. To evaluate the network parameters we use the first split.

5.4.2 Evaluation of LTC network parameters

In the following we first examine the impact of optical flow and data augmentation. We then evaluate gains provided by long-term temporal convolutions for the best flow and data augmentation techniques by comparing 16f and 60f networks. We also investigate the advantage of pre-training on one dataset (UCF101) and fine-tuning on a smaller dataset (HMDB51). Furthermore, we study the effect of systematically increased temporal resolution for flow and RGB inputs as well as the combination of networks.

Optical flow. The impact of the flow quality on action recognition and a comparison to RGB is shown in Figure 5-3 for UCF101 (split 1). The network is trained from scratch and with a 60-frame video volume as input. We first observe that even the low-quality MPEG flow outperforms RGB. The increased quality of optical flow leads to further improvements. The use of Brox flow allows nearly 20% increase in performance. The improvements are consistent when classifying individual clips and full videos. This suggests that action recognition is easier to learn from motion compared to raw pixel values. While results in Figure 5-3 were obtained for 60f networks, the same holds for 16f networks (see Table 5.2). We also conclude that the high accuracy of optical flow estimation plays an important role for learning competitive video representations for action recognition. Given the results in Figure 5-3, we choose Brox flow for all remaining experiments in this work.

Data augmentation. Table 5.1 demonstrates the contribution of data augmentation when training a large CNN with limited amount of data. Our baseline uses sliding window clips with 75% overlap and a dropout of 0.5 during training. We gain 3.1% with random clipping, 1.6% with multiscale cropping and 2% with higher dropout ratio. When combined, the data augmentation and a higher dropout results in a 4% gain for video classification on UCF101 split 1. High dropout, multiscale

Table 5.1 – Data augmentations on UCF101 (split 1). All results are with 60-frame Brox flow and training from scratch. All three modifications (random clipping, multiscale cropping and high dropout) give an improvement when used alone, the best performance is obtained when combined.

Method	Clip accuracy	Video accuracy
Baseline augmentation	71.6	76.5
Random clipping	74.8	79.6
Multiscale cropping	72.5	78.1
High dropout (0.9)	74.4	78.5
Combined	**76.3**	**80.5**

cropping and random clipping are used in the remaining experiments, unless stated otherwise.

Comparison of 16f and 60f networks. Our 16-frame and 60-frame networks have similar complexity in terms of input sizes and the number of network parameters (see Section 5.3). Moreover, the 16-frame network resembles the C3D architecture and enables direct comparison with Tran et al. [2015]. We therefore study the gains provided by the 60-frame inputs before analyzing performance with systematically increasing temporal resolution (from 20 to 100 frames by steps of 20) in the next paragraph.

Table 5.2 compares the performance of 16f and 60f networks for RGB and flow inputs as well as for different data augmentation and dropout ratios for UCF101 split 1. We observe consistent and significant improvement of long-term temporal convolutions in 60f networks for all tested setups, when measured in terms of clip and video accuracies. Our 60f architecture significantly improves for both RGB and flow-based networks. As expected, the improvement is more prominent for clips since video evaluation aggregates information over the whole video.

We repeat similar experiments for the split 1 of HMDB51 and report results in Table 5.3. Similar to UCF101, flow-based networks with long-term temporal convolu-

Table 5.2 – Results for networks with different temporal resolutions and under variation of data augmentation (MS: multiscale cropping) and dropout (D) for UCF101 (split 1), trained from scratch. Random clipping is used in all experiments. Evaluations are on individual clips and on full videos.

Input	MS	D	Test	16f	60f	gain
RGB	x	0.5	Clip	48.4	57.0	+ 8.6
			Video	51.9	59.9	+ 8.0
Flow	x	0.5	Clip	66.8	74.8	+ 8.0
			Video	77.4	79.6	+ 2.2
Flow	✓	0.9	Clip	67.1	**76.3**	+ 9.1
			Video	78.7	**80.5**	+ 1.8

Table 5.3 – Results for networks with different temporal resolutions for HMDB51 (split 1) with or without pre-training on UCF101. Flow input, random clipping, multiscale cropping and 0.9 dropout are used in all setups. We compare our results with 3D CNNs to their 2D CNN counterparts from [Simonyan and Zisserman, 2014]. We note that 3D CNNs outperform 2D CNNs when the temporal resolution is increased.

Pre-training	Test	16f	60f	gain	2D CNN [2014]
x	Clip	37.0	52.6	+15.6	
	Video	43.9	52.9	+ 9.0	46.6
✓	Clip	40.6	**56.1**	+15.5	
	Video	48.3	**57.1**	+ 8.8	49.0

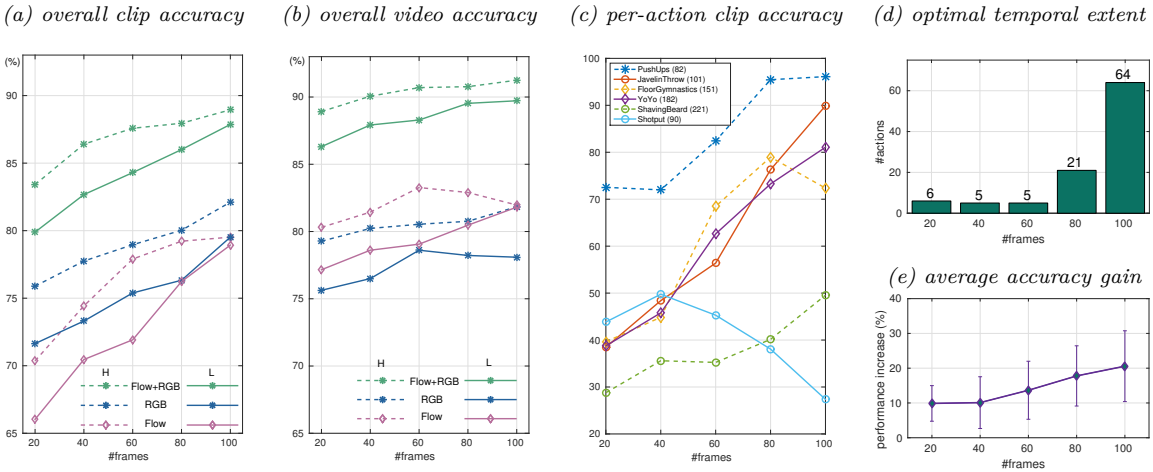


Figure 5-4 – Results for the split 1 of UCF101 using LTC networks of *i.* varying temporal extents t, *ii.* varying spatial resolutions [high (H), low (L)] and *iii.* different input modalities (RGB pre-trained on Sports-1M, flow trained from scratch). For faster convergence all networks were trained using 0.5 dropout and a fixed batch size of 10. Classification results are shown for clips (a) and videos (b) computed over all classes and presented for a subset of individual classes for flow input of low spatial resolution (c). The average number of frames in the training set for a class is denoted in parenthesis. (d) shows a distribution of action classes over the optimal temporal extent and (e) indicates correspondnig improvements (see text for details). With the exception of a few classes, most of the classes benefit from larger temporal extents.

tions lead to significant improvements over the 16f network, in terms of clip and video accuracies. Given the small size of HMDB51, we follow [Simonyan and Zisserman, 2014] and also fine-tune networks that have been pre-trained on UCF101. As illustrated in the 2nd row of Table 5.3, such pre-training gives significant improvement. Moreover, our 60f flow networks significantly outperform results of the 2D CNN temporal stream ([Simonyan and Zisserman, 2014], Table 2) evaluated in a comparable setup, both with and without pre-training.

Varying temporal and spatial resolutions. Given the benefits of long-term temporal convolutions above, it is interesting to study networks for increasing temporal extents and varying spatial resolutions systematically. In particular, we investigate if accuracy saturates for networks with larger temporal extents, if higher spatial resolution impacts the performance of long-term temporal convolutions and if LTC is equally beneficial for flow and RGB networks.

To study these questions, we evaluate networks with increasing temporal extent $t \in \{20, 40, 60, 80, 100\}$ and two spatial resolutions $\{58 \times 58, 71 \times 71\}$ for both RGB and flow. We also investigate combining RGB and flow by averaging their class scores. Preliminary experiments with alternative fusion techniques did not improve over such a late fusion.

Flow networks have our previous architecture as in Figure 5-2, except slightly more connections in *fc6* for 71×71 resolution. For flow input, we train our networks from scratch. For RGB input, learning appears to be difficult from scratch. Even if we extend the temporal extent from 60 frames (see Table 5.2) to 100 frames, we obtain 68.4% on UCF101 split 1, which is still below frame-based 2D convolution methods fine-tuned from ImageNet pre-training [Simonyan and Zisserman, 2014]. Although longer extent boosts the performance significantly, we conclude that one needs to pre-train RGB network on larger data.

Given the large improvements provided by the pre-training of C3D RGB network

on the large-scale Sports-1M dataset in [Tran et al., 2015], we use this 16-frame pre-trained network and extend it to longer temporal convolutions in 2 steps.[1] The first step is fine-tuning the 16f C3D network. A randomly initialized fully connected (*fc*) layer of size 101 (number of classes) is added at the end of the network. Only the *fc* layers are fine-tuned by freezing the convolutional layers. We start with a learning rate of 3×10^{-4} and decrease it to 3×10^{-5} after 30K iterations for 1K more iterations. In the second step, we input longer clips to the network and fine-tune all the layers. Convolutional layers are applied to longer video clips of t frames. This results in outputs from *conv5* layer with $\lfloor t/16 \rfloor$ temporal resolution. To re-cycle pre-trained *fc* layers of C3D, we max-pool *conv5* outputs over time and pass results to *fc6*. We use a subset of the *fc6* weights for inputs of lower spatial resolution. For this phase, we run for same number of iterations, but we decrease the learning rate from 3×10^{-5} to 3×10^{-6}. We keep dropout ratio 0.5 as in the pre-trained network.

Figure 5-4(a)(b) illustrates results of networks with varying temporal and spatial resolutions for clips and videos of UCF101, split 1. We observe significant improvements over t for LTC networks using flow (trained from scratch), RGB (with pre-training on Sports-1M), as well as combination of both modalities. Networks with higher spatial resolutions give better results for lower t, however, the gain of increased spatial resolution is lower for networks with long temporal extents. Given the large number of parameters in high-resolution networks, such behavior can be explained by the overfitting due to the insufficient amount of training data in UCF101. We believe that larger training sets could lead to further improvements. Moreover, flow benefits more from the averaging of clip scores than RGB. This could be an indication of static RGB information over different time intervals of the video, whereas flow is dynamic.

Figure 5-4(c) presents results of LTC for a few action classes demonstrating a variety of accuracy patterns over different temporal extents. Out of all 101 classes,

[1] We have also tried to pre-train our flow-based networks on Sports-1M but did not obtain significant improvements.

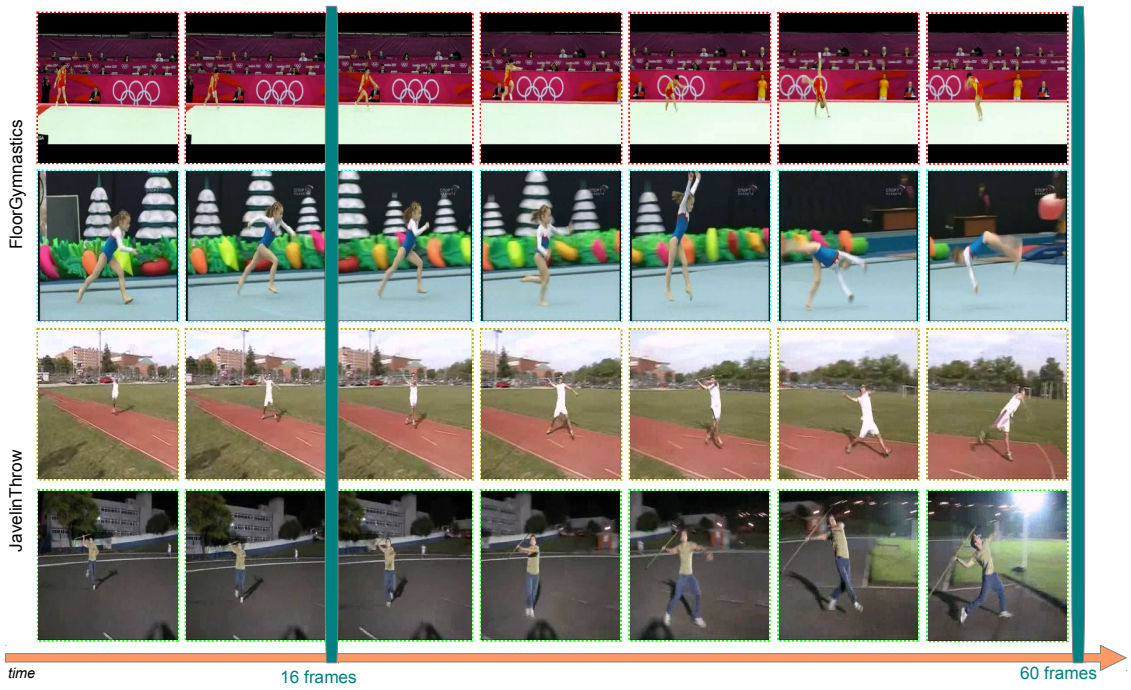


Figure 5-5 – The highest improvement of long-term temporal convolutions in terms of class accuracy is for *JavelinThrow*. For 16-frame network, it is mostly confused with the *FloorGymnastics* class. Here, we visualize sample videos with 7 frames extracted at every 8 frames. The intuitive explanation is that both classes start by running for a few seconds and then the actual action takes place. LTC can capture this interval, whereas 16-frame networks fail to recognize such long-term activities.

no action has monotonic decrease with the increasing temporal extent, whereas the performance of 25 action classes increased monotonically. *PushUps*, *YoYo* and *ShavingBeard* are examples of classes with high, medium and low performance that all benefit from larger temporal extents. *Shotput* is an example of a class with lower performance for longer temporal extents. A possible explanation is that samples of the *Shotput* class are relatively short and have 90 frames on average (we pad short clips). Two additional examples with a significant gain for larger temporal extents are *FloorGymnastics* and *JavelinThrow*, see Figure 5-5 for sample frames from these two classes. We observe that both actions are composed of running followed by throwing a javelin or the actual gymnastics action. Short-term networks, thus, easily confuse the two actions, while LTC can capture such long and complex actions. For both classes, we provide snapshots at every 8th frame. It is clear that one needs to look at more than 16 frames to distinguish these actions.

Let the performance of class c for temporal extent t be $p_c(t)$. A set of classes with the maximum performance at t is then $M(t) := \{c \mid t \in \arg\max_{t'}(p_c(t'))\}$. Figure 5-4(d) plots $|M(t)|$ with respect to t. The majority of classes (64 out of 101) obtain maximum performance when trained with 100f networks. To further check if there exists an "ideal temporal extent" for different actions, Figure 5-4(e) illustrates the average performance increase $d(t)$:

$$d(t) := \frac{1}{|M(t)|} \sum_{M(t)} \max_{t'}(p_c(t')) - \min_{t'}(p_c(t')) \tag{5.1}$$

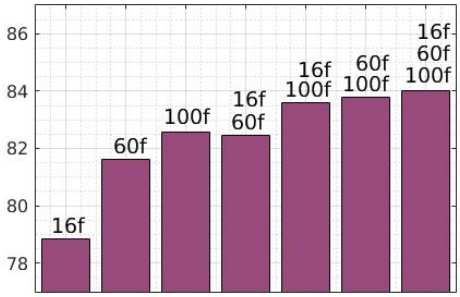
We can observe that values of $d(t)$ are lower for shorter extents and larger for longer extents. That means actions scoring best at short extents score similar at all scales, so we cannot conclude that certain actions favor certain extents. Most actions favor long extents as the difference is largest for 100f. A possible explanation is that making the interval too long for short actions does not have much impact, whereas making the interval too short for long actions does impact the performance, see Figure 5-5.

Combining networks of varying temporal resolutions. We evaluate combining different networks with late fusion. For final results on flow, 58×58 spatial resolution and 0.9 dropout are used for both UCF101 and HMDB51 datasets. The flow networks are learned from scratch for UCF101 and fine-tuned for HMDB51. For final results on UCF101 with RGB input, we use 71×71 spatial resolution networks fine-tuned from C3D network [Tran et al., 2015]. However, we do not further fine-tune it for HMDB51 because of overfitting, and use C3D network as a feature extractor in combination with SVM for obtaining RGB scores. Our implementation of C3D as a feature extractor and a SVM classifier achieved 80.2% and 49.7% average performance on 3 splits of UCF101 and HMDB51, respectively. We get similar result when fine-tuning C3D on 16-frames (80.5% on UCF101).

Figure 5-6 (left) shows results for combining outputs of flow networks with different temporal extents. The combination is performed by averaging video-level class scores produced by each network. We observe that combinations of two networks with different temporal extents provides significant improvement for flow. The gains of combining more than two resolutions appear to be marginal. For final results, we report combining 60f and 100f networks for both flow and RGB, with the exception of HMDB51 RGB scores for which we use a SVM classifier on 16f feature extractor. Figure 5-6 (right) shows results for combining multiscale networks of different modalities together with the IDT+FV baseline classifier [Wang and Schmid, 2013] on split 1 of both datasets. We observe complementarity of different networks and IDT+FV where the best result is obtained by combining all classifiers.

5.4.3 Comparison with the state of the art

In Table 5.4, we compare to the state of the art on HMDB51 and UCF101 datasets. Note that the numbers do not directly match with previous tables and figures, which are reported only on first splits. Different methods are grouped together according to being hand-crafted, using only RGB or optical flow input to CNNs and combining



	UCF101	HMDB51
LTC_{Flow} (100f)	82.6	56.7
LTC_{Flow} (60f+100f)	83.8	60.5
LTC_{RGB} (100f)	81.8	-
LTC_{RGB} (60f+100f)	81.5	-
$\text{LTC}_{Flow+RGB}$	91.0	65.6
$\text{LTC}_{Flow+RGB}$+IDT	**91.8**	**67.7**

Figure 5-6 – Results for network combinations. (Left): Combination of LTC flow networks with different temporal extents on UCF101-split 1. (Right): Combination of flow and RGB networks together with IDT features on UCF101 and HMDB51-splits 1. For UCF101, flow is trained from scratch and RGB is pre-trained on Sports-1M. For HMDB51, flow is pre-trained on UCF101 and RGB scores are obtained using C3D feature extractor.

any of these. Trajectory features perform already well, especially with higher-order encodings. CNNs on RGB perform very poor if trained from scratch, but strongly benefits from static image pre-training such as ImageNet. Recently [Tran et al., 2015] trained space-time filters from a large collection of videos; however, their method is not end-to-end, given that one has to train a SVM on top of the CNN features. Although we fine-tune LTC_{RGB} based on a network learned with a short temporal span and we reduce the spatial resolution, we are able to improve by 2.2% on UCF101 (80.2% versus 82.4%) by extending the pre-trained network to 100 frames.

We observe that LTC outperforms 2D convolutions on both datasets. Moreover, LTC_{Flow} outperforms LTC_{RGB} despite no pre-training. Our results using $\text{LTC}_{Flow+RGB}$ with average fusion significantly outperform the Two-stream average fusion baseline [Simonyan and Zisserman, 2014] by 4.8% and 6.8% on UCF101 and HMDB51 datasets, respectively. Moreover, the SVM fusion baseline in [Simonyan and Zisserman, 2014] is still significantly below $\text{LTC}_{Flow+RGB}$. Overall, the combination of our best networks $\text{LTC}_{Flow+RGB}$ together with the IDT method[2] provides best results on both UCF101 (92.7%) and HMDB51 (67.2%) datasets. Notably both of these

[2]Our implementation of IDT+FV [Wang and Schmid, 2013] obtained 84.5% and 57.3% for UCF101 and HMDB51, respectively.

		Method	UCF101	HMDB51
IDT	[Wang and Schmid, 2013]	IDT+FV	85.9	57.2
	[Lan et al., 2015]	IDT+MIFS	89.1	65.1
RGB	[Karpathy et al., 2014]	Slow fusion (from scratch)	41.3	-
	[Tran et al., 2015]	C3D (from scratch)	44[1]	-
	[Karpathy et al., 2014]	Slow fusion	65.4	-
	[Simonyan and Zisserman, 2014]	Spatial stream	73.0	40.5
	[Tran et al., 2015]	C3D (1 net)	82.3	-
	[Tran et al., 2015]	C3D (3 nets)	85.2	-
Flow	[Simonyan and Zisserman, 2014]	Temporal stream	83.7	54.6
RGB + Flow	[Simonyan and Zisserman, 2014]	Two-stream (avg. fusion)	86.9	58.0
	[Simonyan and Zisserman, 2014]	Two-stream (SVM fusion)	88.0	59.4
	[Ng et al., 2015]	LSTM	88.6	-
	[Wang et al., 2015a]	TDD	90.3	63.2
	[Wang et al., 2016b]	Transformations	92.4	62.0
	[Feichtenhofer et al., 2016]	Two-stream (conv. fusion)	**92.5**	**65.4**
+IDT	Tran et al. [2015]	C3D+IDT	90.4	-
	[Wang et al., 2015a]	TDD+IDT	91.5	65.9
	[Feichtenhofer et al., 2016]	Two-str. (conv. fusion)+IDT	**93.5**	**69.2**
		LTC_{RGB}	82.4	-[2]
		LTC_{Flow}	85.2	59.0
		$\text{LTC}_{Flow+RGB}$	**91.7**	**64.8**
		$\text{LTC}_{Flow+RGB}$+IDT	**92.7**	**67.2**

Table 5.4 – Comparison with the state of the art on UCF101 and HMDB51 (mean accuracy across 3 splits). [1]This number is read from the plot in figure 2 [Tran et al., 2015] and is clip-based, therefore not directly comparable. [2]We use C3D+SVM scores (49.7%) for HMDB51.

results outperform previously published results on these datasets, except [Feichtenhofer et al., 2016] which studies best ways to combine RGB and flow streams, hence complementary to our method.

5.4.4 Analysis of the 3D spatio-temporal filters

First layer weights. In order to have an intuition of what an LTC network learns, we visualize the first layer space-time convolutional filters in the vector-field form. Filters learned on 2-channel optical flow vectors have dimension $2 \times 3 \times 3 \times 3$ in

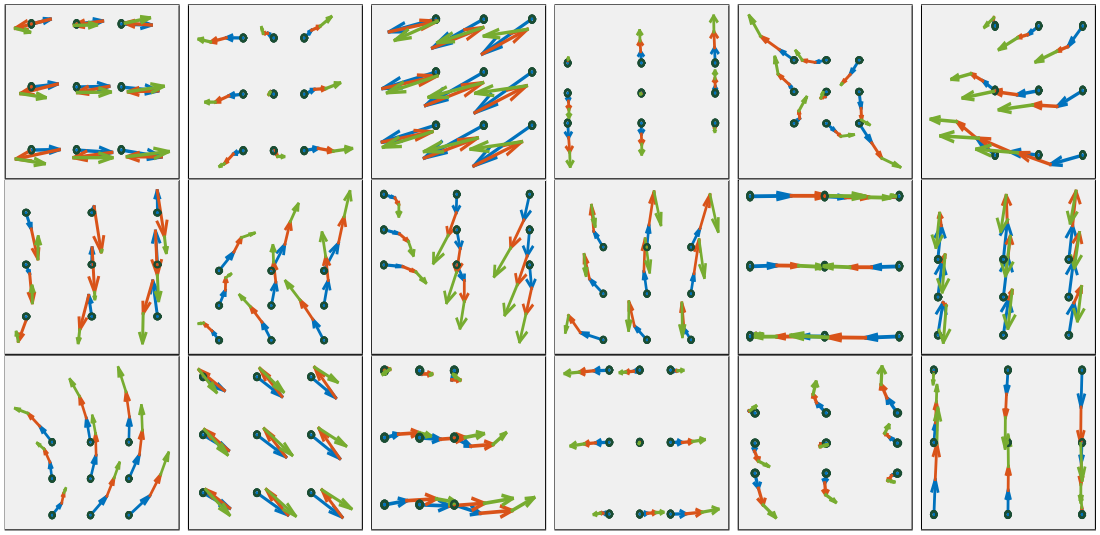


Figure 5-7 – Spatio-temporal filters from the first layer of the network learned with 2-channel, Brox optical flow and 60 frames on UCF101. 18 out of 64 filters are presented. Each cell in the grid represents two $3 \times 3 \times 3$ filters for 2-channel flow input (one for x and one for y). x and y intensities are converted into vectors in 2D. Third dimension (time) is denoted by putting vectors one after the other in different colors for better visualization (t=1 blue, t=2 red, t=3 green). We see that LTC is able to learn complex motion patterns for video representation. Better viewed in color.

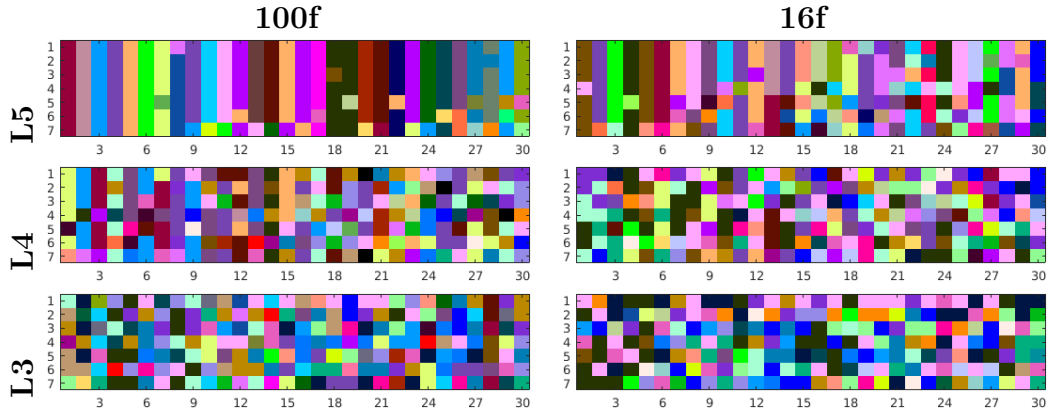
terms of channels, width, height and time. For each filter, we take the two channels in each $3 \times 3 \times 3$ volume and visualize them as vectors using x- and y-components. Figure 5-7 shows 18 example filters out of the 64 from a network learned on UCF101 with 60 frames flow input. Since our filters are spatio-temporal, they have a third dimension in time. We find it convenient to show them as vectors concatenated one after the other with regard to the time steps. We denote each time step with different colors and see that the filters learned by long-term temporal convolutions are able to represent complex motions in local neighborhoods, which enables to incorporate even more complex patterns in later stages of the network.

High-layer filter activations. We further investigate filters from higher convolutional layers by examining their highest activations. For a given layer and a chosen

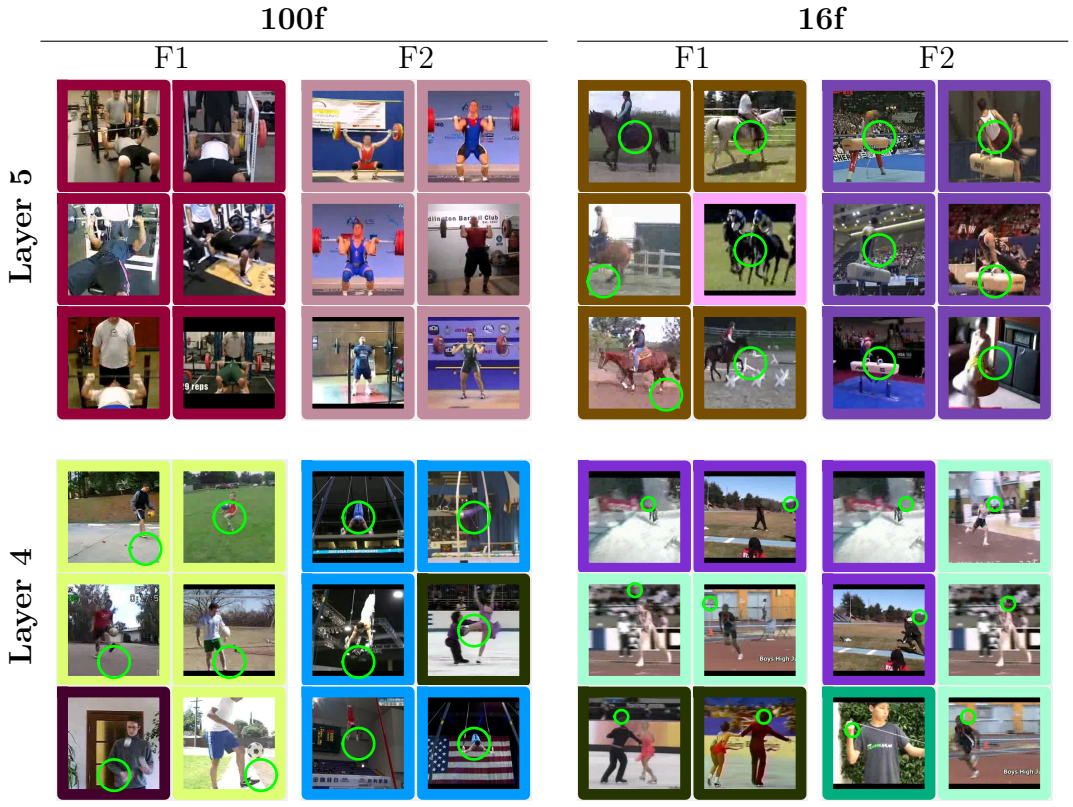
filter, we record the maximum activation value for all test videos[3] for that filter. We then sort test videos according to the activation values and select the top 7 videos. This procedure is similar to [Zeiler and Fergus, 2014]. We can expect that a filter should be activated by similar action classes especially at the higher network layers. Given longer video clips available to the LTC networks, we also expect better grouping of actions from the same class by filter activations of LTC. We illustrate action classes for 30 filters (x-axis) and their top 7 activations (y-axis) for the 100f and 16f networks in Figure 5-8(a). Each action class is represented by a unique color. The filters are sorted by their purity, i.e. the frequency of the dominating class. We assign each video the color of its ground truth class label. We see that the clustering of videos from the same class becomes more clear in higher layers in the network for both 16f and 100f networks. However, it is evident that 100f filters have more purity than 16f even in L4 and L3. Note that 16f network is trained with high resolution (112×112) flow and 100f network with low resolution (58×58) flow.

Example frames from top-scoring videos for a set of selected filters f are shown in Figure 5-8(b) for 16f and 100f flow networks. We also provide a video on our project web page [LTC project page] to show which videos activate for these filters. We can observe that for filters f maximizing the homogeneity of returned class labels, the top activations for filters of the 100f network result in videos with similar action classes. The grouping of videos by classes is less prominent for activations of the 16f network. This result indicates that the LTC networks have higher level of abstraction at corresponding convolution layers when compared to networks with smaller temporal extents.

[3]UCF101 videos are obtained by clipping different parts (*video*) from a longer video (*group*). We take one *video* per *group* assuming that *videos* from the same *group* would have similar activations and would avoid a proper analysis. In total, there are 7 test *groups* per class; therefore there can be at most 7 *videos* belonging to a class.



(a) Top activations of filters at *conv3-conv5* layers. Each row is another layer, indicated by L3-L5. Left is for 100 frames and right is for 16 frames networks. Colors indicate different action classes. Each color plot illustrates distribution of classes for seven top activations of 30 selected filters. Rows are maximum responding test videos and columns are filters.



(b) Frames corresponding to videos with top activations at *conv4* and *conv5* layers. Circles indicate the spatial location of the maximum response. The visualized frames correspond to the maximum response in time.

Figure 5-8 – Comparison of 100f and 16f networks by looking at the top activations of filters. Better viewed in color.

5.4.5 Runtime

Training on UCF101 takes 1.9 day for 100f (58x58) networks and 1.1 day for 16f (112x112) networks with 0.5 dropout. At test time (without flow computation), the 100f and 16f networks run at 4452fps and 1128fps respectively on a Titan X GPU and 8 CPU cores for parallel data loading. Although it takes more time (roughly 1.6 times) to compute one forward pass for 100f, a larger number of frames are processed per second. C3D [Tran et al., 2015] reports 313fps for a 16f network while using a larger number of parameters. Our proposed solution is therefore computationally efficient.

5.5 Conclusions

This chapter introduced and evaluated long-term temporal convolutions (LTC) and showed that they can significantly improve the performance. Using space-time convolutions over a large number of video frames, we obtained state-of-the-art performance on two action recognition datasets UCF101 and HMDB51. We also demonstrated the impact of the optical flow quality. In the presence of limited training data, using flow improves over RGB and the quality of the flow impacts the results significantly.

Here, we have focused on the discriminative power of the 3D CNN representations for action recognition. In the next chapter, we will focus on the view-independence property of the 3D CNN representations.

Chapter 6

View-independent Video Representations for Action Recognition

This chapter presents our second contribution on action recognition. We address the problem of recognizing human actions given RGB videos of unseen viewpoints. Typical methods using convolutional neural networks do not explicitly handle viewpoint variation, instead rely on diverse training data. We carefully design and analyze challenging cross-view action recognition settings and observe that a naive training approach poorly generalizes to unseen views. We propose a simple yet effective similarity training that introduces additional constraints such that synchronous videos of different views are closer in the representation space. This light-weight supervision can be seen as a generic regularization that can be integrated in various settings. In the presence of a single view available for training, we show that this similarity can be learned on a separate multi-view dataset. Our experimental study on N-UCLA, NTU RGB+D, and UESTC multi-view action recognition datasets shows the effectiveness of the similarity training.

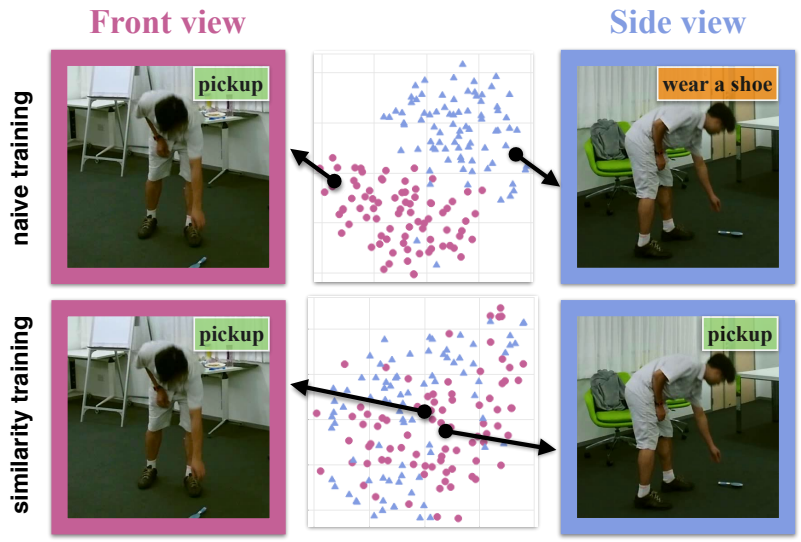


Figure 6-1 – State-of-the-art models trained only on front view generalizes poorly on side view videos (top). The t-SNE visualization confirms the clear separation of the representation space within views. Our training with similarity constraints improves the generalization to unseen viewpoints (bottom). Green and orange labels denote correct and incorrect action predictions, respectively.

6.1 Introduction

Learning human action representations from RGB video data has been widely studied. Recent advances on convolutional neural networks (CNNs) have shown excellent performance [Carreira and Zisserman, 2017; Feichtenhofer et al., 2016; Hara et al., 2018] on benchmark datasets, such as UCF101 [Soomro et al., 2012]. We take a state-of-the-art action recognition network [Hara et al., 2018] and train it on videos from a benchmark dataset NTU RGB+D [Shahroudy et al., 2016] using the videos where people are facing the camera. When we test this network again on front-view videos, we obtain 80% accuracy. When we test with side-view videos, the performance drops to 40%. This effect is illustrated in Figure 6-1.

As also confirmed by Liu et al. [2011b]; Zheng et al. [2016], the appearance of an action is drastically different when performed from different viewpoints. This might be a desired property for tasks such as human pose estimation. However, action

recognition approaches need to be view-invariant. For surveillance, placing a new camera in the environment would otherwise require re-training a model. Similarly, an ambient assisted living system would not generalize from one environment to another. Overall, for action recognition, generalization is important when viewpoints vary.

Achieving view-invariance is mainly challenging because the training datasets have a strong bias towards cameras capturing from a certain view. A naive way to solve this problem is to collect data from all possible views, but this is impractical. CNN architectures are not designed to be view-invariant since they operate on 2D image projections of the 3D world. Thus, we argue that additional supervision is necessary to ensure similarity between representations of different views.

Existing methods addressing the cross-view action recognition problem do not work in challenging setups. The most popular large-scale multi-view action recognition dataset to date is NTU RGB+D [Shahroudy et al., 2016] and the cross-view protocol has the same subjects both in training and test. Moreover, in the standard protocol, the training is performed on 0° and 90° views, while the test is on 45°. Hence, the state of the art is close to saturation obtaining over 90% accuracy. In practice, this problem is not close to be solved (see Figure 6-1). Therefore, we introduce and study more challenging setups.

Cross-view action recognition has mostly been studied based on skeleton data as input [Kong et al., 2017; Liu et al., 2017b; Zhang et al., 2017]. Transforming RGB features to be view-invariant is not as trivial as transforming 3D skeletons which already encode the view information explicitly. Recent work of Wang et al. [2018a] focuses on view-specific RGB features to achieve cross-view recognition. They introduce a multi-branch neural network that trains one branch per view, as well as a view classification. The view classification scores are then used as weights when fusing view-specific branches. While achieving high performance, this approach is costly and it cannot handle single-view training.

In this work, we investigate adapting multi-frame video representations to become

view-invariant. Our model uses only RGB input, therefore does not rely on other modalities such as skeleton or depth. We first introduce challenging protocols and emphasize the difficulty of the problem. Then, we study a similarity loss whose goal is to bring representations closer in feature space if the videos are captured temporally synchronously from different views. Furthermore, we extend to scenarios where only one training view is available with action annotations. For this case, we demonstrate that the similarity loss can be optimized on a separate multi-view dataset, while the action loss is optimized on the single-view dataset. Our contribution is two-fold: (i) we integrate a conceptually simple, but powerful similarity supervision into the training of 3D CNNs for cross-view action recognition; (ii) we systematically study and show the benefits of such approach on three benchmark datasets. Our implementation and trained models will be made available publicly.

6.2 Related Work

Action recognition is a well-established research field. For a broad review of the literature on action recognition, see the recent survey of Kong and Fu [2018] and our literature review in Chapter 2, Section 2.2. Here, we focus on relevant works on cross-view action recognition and the use of similarity training in other domains.

Cross-view action recognition. Due to the difficulty of building cross-view action recognition datasets, the standard benchmarks have been recorded in controlled environments. RGB-D datasets such as IXMAS [Weinland et al., 2007], UWA3D II [Rahmani et al., 2016] and N-UCLA [Wang et al., 2014] were state-of-the-art evaluation until the availability of the large-scale NTU RGB+D dataset Shahroudy et al. [2016]. The size of the NTU dataset allowed training and evaluating neural network approaches unlike previous datasets that were medium-sized. Very recently, Ji et al. [2018] collected the first large-scale dataset that has a 360° coverage around the performer, although still in a lab setting.

Since multi-view action datasets are typically captured with depth sensing devices, such as Kinect, they also provide an accurate estimate of the 3D skeleton. Skeleton-based cross-view action recognition therefore received a lot of attention in the past decade [Ke et al., 2017; Liu et al., 2016, 2017a,b; Zhang et al., 2017]. Variants of LSTMs have been widely used [Liu et al., 2016, 2017a; Shahroudy et al., 2016]. Recently, spatio-temporal skeletons were represented as images [Ke et al., 2017] or higher dimensional objects [Liu et al., 2017b] where standard CNN architectures were applied.

RGB-based cross-view action recognition is in comparison less studied. Early work on transferring appearance features from the source view to the target view explored the use of maximum margin clustering to build a joint codebook for temporally synchronous videos [Farhadi and Tabrizi, 2008]. Following this approach, several other works focused on building global codebooks to extract view-invariant representations [Kong et al., 2017; Rahmani et al., 2018; Zheng and Jiang, 2013; Zheng et al., 2016]. Recently, end-to-end approaches used human pose information as guidance for action recognition [Baradel et al., 2017; Liu and Yuan, 2018; Luvizon et al., 2018; Zolfaghari et al., 2017]. Baradel et al. [2018] introduced glimpse clouds, an attention mechanism that exploits generic 2D interest points. Luo et al. [2018] investigated the use of multi-modal data at training while testing with a single modality. They introduced graph distillation within modalities such as RGB, depth, skeleton, and optical flow. Wang et al. [2018a] proposed to fuse view-specific features from a multi-branch CNN.

Our work differs from previous action recognition methods as follows. We learn a common representation space for temporally synchronous RGB videos in an end-to-end manner by integrating a similarity loss. This approach is light-weight and does not involve any extra cost at test time unlike expensive methods such as [Wang et al., 2018a]. This auxiliary supervision can be applied in a cross-dataset setting if only single-view videos are available with action labels.

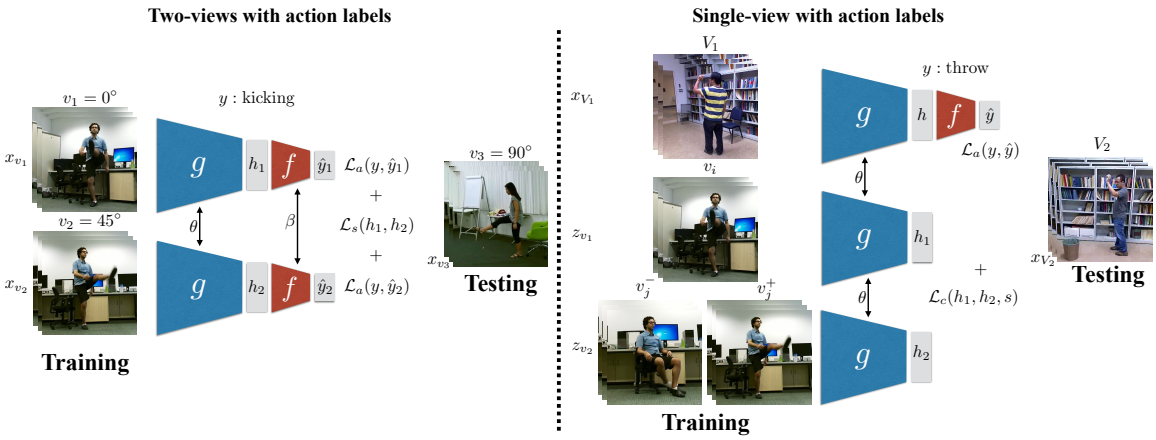


Figure 6-2 – We present an overview of our approach for two different scenarios. Left: The first scenario depicts multi-view action recognition training ($\mathcal{L}_a$) with similarity constraints on the video representation ($\mathcal{L}_s$). Sample images are shown from the NTU dataset to illustrate training with 0° and 45° viewpoints, and testing with a 90° view of different subjects. Right: In the case of single view available at training, we use videos from a different multi-view dataset and integrate a contrastive loss $\mathcal{L}_c$ that requires both positive (x_j^+) and negative (x_j^-) samples. In both cases, the parameter θ of the 3D CNN g is shared among views, thus achieving view-invariance for action recognition. V_1, V_2 represent views from the N-UCLA dataset, whereas v_1, v_2, v_3 denote the views of the NTU dataset. See Section 6.3 for details on the method description as well as the notation.

Similarity learning in other contexts. Representations learned by CNNs can be constrained or adapted for various purposes. Similarity training has been mostly used in the context of metric learning [Yi et al., 2016; Zagoruyko and Komodakis, 2015]. Siamese networks [Bromley et al., 1993] are widely adopted to maximize the similarity between pairs of inputs. Alternatively, several works [Sermanet et al., 2018; Sigurdsson et al., 2018] explore triplets where one positive and one negative input is sampled per anchor input. The goal then becomes minimizing and maximizing the distances to the positive and negative samples, respectively.

In the context of multi-view similarity training, Subramaniam et al. [2016] improve person re-identification by enforcing the image representations of different viewpoints to be closer. Kan et al. [2016] propose a multi-view deep CNN for the generic problem of aligning multiple modalities to a common space. They compare their approach with CCA-based methods. Vo and Hays [2016] aim to match ground-level and satellite images using variants of Siamese CNNs. Time contrastive networks [Sermanet et al., 2018] and their multi-frame follow-up [Dwibedi et al., 2018] explore self-supervised feature learning given multi-view videos in robotics context. Cao et al. [2018] adapt image features of profile faces to a frontal canonical pose for face recognition. Matching deep representations of different views has been observed to be beneficial for these tasks, but it has not been explored for video-based cross-view action recognition.

Our work is also related to domain adaptation methods. Similarity training has been used to adapt the synthetic domain features to the real domain [Massa et al., 2016; Rozantsev et al., 2018]. Inversely, Rad et al. [2018] learn an adaptation layer to map the real image domain to the synthetic domain by jointly optimizing for the target task and a similarity for the adapted features. They show improvements on 3D hand pose and object pose problems with this approach. Wang and Hebert [2016] transform features learned on small sample size regime to larger collections with the aim of learning a novel category or for one-shot recognition. Similar to our method, Sigurdsson et al. [2018] improve the action recognition performance by defining a

triplet similarity loss for the first- and third-person videos, but their work focuses more on the semantic alignment of object interactions rather than a geometric alignment. Moreover, unlike [Sigurdsson et al., 2018], our work aims at aligning multiple-frame video representations.

6.3 View-invariant action representations

In the following, we describe our similarity training approach for two scenarios illustrated in Figure 6-2. We then give details on the implementation.

6.3.1 Cross-view similarity training

Following the success of 3D CNNs for video recognition [Carreira and Zisserman, 2017; Hara et al., 2018; Tran et al., 2015; Varol et al., 2018b], we employ a spatio-temporal convolutional architecture that operates on multi-frame video inputs. Formally, an RGB video input $x \in \mathbb{R}^{t \times w \times h \times 3}$ of t frames and of $w \times h$ spatial resolution is mapped to a feature representation $g_\theta(x) \in \mathbb{R}^d$ by the 3D CNN architecture g parameterized by θ. To provide an initial representation, we train our networks with conventional action classification supervision, namely cross-entropy loss defined as:

$$\mathcal{L}_a(y, p) = -\sum_{i}^{C} y_i \log(p_i), \tag{6.1}$$

where C is the total number of action classes, $p \in \mathbb{R}^C$ is the predicted probability distribution output by the network, and y is the one-hot encoding of the ground truth label. Given a dataset of N labeled videos $\{x^j, y^j\}_j^N$, our goal is first to minimize $\mathcal{L}(y, f_\beta \circ g_\theta(x))$ over the training samples. Here, f denotes the classification function given the feature representation. This optimization gives an initial estimate for the network parameters β and θ, which we call the baseline model.

The baseline model is not robust against viewpoint changes, meaning the feature

representation $g_\theta(x_{v_1})$ and $g_\theta(x_{v_2})$ for two synchronized videos (x_{v_1}, x_{v_2}) from different viewpoints (v_1, v_2) are not guaranteed to be similar. With this motivation, we enforce additional constraints on the distance between feature representations of synchronized videos. Our similarity supervision does not require modifying the baseline architecture, but requires updating its parameters.

We study two scenarios for the similarity training. The first one is when we have access to synchronous video recordings together with their action label. The second case is when we have a single viewpoint available with action annotations and other multi-view videos without labels.

Two-views with action labels. Given two temporally synchronized multi-view video inputs (x_{v_1}, x_{v_2}) and their action label y, our goal is to minimize:

$$\begin{aligned}\mathcal{L}_a(y, f_\beta \circ g_\theta(x_{v_1})) + \mathcal{L}_a(y, f_\beta \circ g_\theta(x_{v_2})) \\ + \mathcal{L}_s(g_\theta(x_{v_1}), g_\theta(x_{v_2})),\end{aligned} \tag{6.2}$$

where the last term $\mathcal{L}_s(h_1, h_2) = \|h_1 - h_2\|_2^2$ regularizes the latent representation to bring multi-view video features closer in Euclidean space. Note that the parameters are shared between the two views. With this optimization, we adapt network parameters β and θ such that g_θ function becomes view-invariant. In practice, if we have more than two training viewpoints, we sample all possible (x_{v_p}, x_{v_q}) pairs.

Single-view with action labels. A more general and challenging scenario is to have action labels only for a single view. Our aim in this case is to adapt the representations g_θ to achieve view-invariance with the help of other multi-view video sources. Let $\{z^j_{v_1}, z^j_{v_2}\}_j^M$ be a separate dataset of M paired videos of two different views. We then seek to minimize:

$$\mathcal{L}_a(y, f_\beta \circ g_\theta(x)) + \mathcal{L}_s(g_\theta(z_{v_1}), g_\theta(z_{v_2})). \tag{6.3}$$

Here, the input domain x is different from the input domain z. Therefore, the distance objective can be trivially minimized by outputting zero for the input z. To alleviate this issue, we use a contrastive loss that also maximizes the distance from negative samples by a margin m:

$$\begin{aligned}\mathcal{L}_c(h_1, h_2, s) = s\|h_1 - h_2\|_2^2 \\ + (1-s)\max(0, m^2 - \|h_1 - h_2\|_2^2),\end{aligned} \tag{6.4}$$

where the positivity $s = 1$ when (h_1, h_2) are feature representations of temporally synchronous videos and $s = 0$ otherwise. Hence, the final cross-dataset optimization objective becomes:

$$\mathcal{L}_a(y, f_\beta \circ g_\theta(x)) + \mathcal{L}_c(g_\theta(z_{v_1}), g_\theta(z_{v_2})). \tag{6.5}$$

In our experiments, we show that in both cases, the additional regularization on the representation space improves the action recognition performance, especially in cross-view experiments when the test is performed on videos of unseen views at training. This shows that by enforcing vicinity between views v_1 and v_2, we implicitly reduce the distance to an arbitrary unseen view v_3, hence become view-invariant.

6.3.2 Implementation details

As illustrated in Figure 6-2, we have a shared network g, which is a 3D ResNet50 [Hara et al., 2018] obtained by inflating the weights of a 2D ResNet50 [He et al., 2015] and pre-training on the Kinetics dataset [Kay et al., 2017]. At the end of the last residual connection, the feature responses are spatio-temporally averaged to form a 2048-dimensional representation, which becomes the output of the function g. Note that we remove the last ReLU layer of ResNet50 not to limit the information capacity of the features. The action classification function f is a fully connected layer followed by

a softmax. The input videos have 16 frames and 256×256 spatial resolution. During training, we randomly sample 16 consecutive frames. At test time, we use a sliding window without overlapping and average the softmax scores.

6.4 Experiments

In this section, we start by presenting the three action recognition datasets used in our experiments (Section 6.4.1). We first conduct an analysis on a challenging setup and show an ablation for different components of our method (Section 6.4.2). Next, we experiment with the single-view scenario (Section 6.4.3). Then, we compare our results with the state-of-the-art methods on benchmarks (Section 6.4.4). Finally, we show a qualitative analysis (Section 6.4.5).

6.4.1 Datasets and evaluation

We briefly present the three datasets used in this work as well as the evaluation protocols employed. Figure 6-3 shows the viewpoints available in each of the datasets.

N-UCLA multi-view action 3D dataset (N-UCLA). This dataset [Wang et al., 2014] consists of 1475 videos divided into 3 views from 10 actions. There are missing videos and the temporal synchronization among views is imperfect. Each sequence is 20-frames long on average. The standard protocol [Wang et al., 2014] measures accuracy when training on all possible view pairs and testing on the remaining view. $V^k_{i,j}$ denotes training with views (i, j) and testing with view k. In addition to the standard protocol, we also report for the single-view training setup which is more challenging. This dataset is relatively small for training deep neural networks. Similar to [Baradel et al., 2018], we pre-train our networks on NTU. We use this dataset to illustrate the case of single-view training data with action labels by performing cross-dataset experiments.

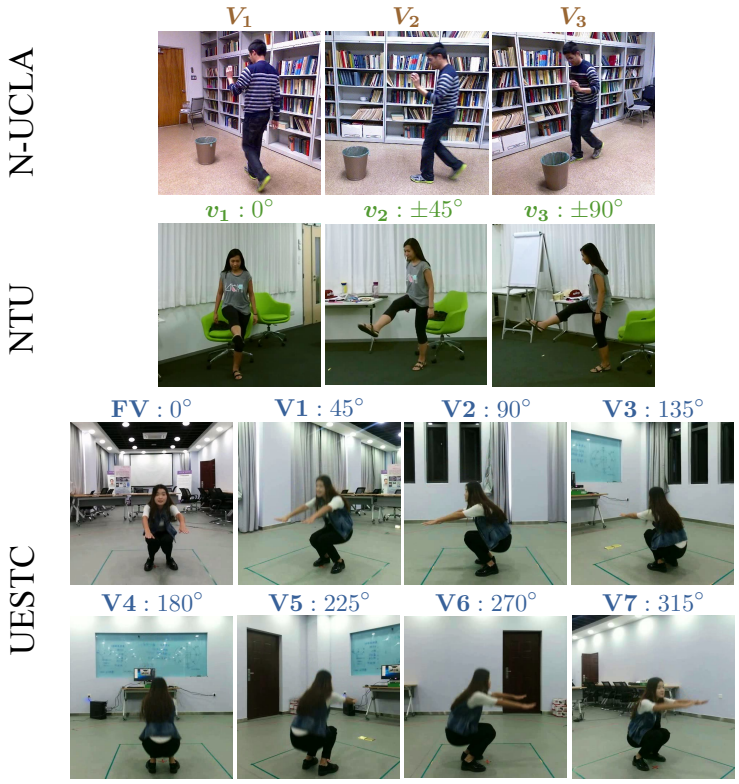


Figure 6-3 – We show multi-view samples from the datasets used. N-UCLA and NTU datasets have 3 different views. N-UCLA views are not well separated, but NTU views correspond to 0°, 45°, and 90° from left to right. UESTC has a very clear separation of views with 8 views covering 360° around the performer.

NTU RGB+D dataset (NTU). This dataset [Shahroudy et al., 2016] captures 60 actions with 3 synchronous cameras. The large scale (56K videos) of the dataset allows training deep neural networks. Each sequence has 84 frames on average. The standard protocols [Shahroudy et al., 2016] report accuracy for cross-view and cross-subject splits. The cross-view split considers 0° (v_1) and 90° (v_3) views as training and 45° (v_2) view as test, and the same subjects appear both in training and test. Note that the official camera enumeration (C1, C2, C3) [Shahroudy et al., 2016] does not correspond to our view enumeration. We select v_1 as the first replications with C3 and the second replications with C2. Similarly, v_3 denotes videos from the first replications with C2 and the second replications with C3. v_2 corresponds to C1. This is because each subject replicates the action once facing camera 2 and another time facing camera 3 in a systematic way. We report on the standard protocols to be able to compare to the state of the art. However, we introduce new protocols to make the task more challenging and to be able to see the effect of view-invariant features. From the cross-subject training split, we separate the 0° and 45° views for training and we test on the 0°, 45°, 90° views of the cross-subject test split. We call this protocol cross-view-subject. Our focus is mainly to improve the unseen view of 90°.

UESTC RGB-D varying-view 3D action dataset (UESTC). UESTC is a recent dataset [Ji et al., 2018] that systematically collects 8 equally separated viewpoints that cover 360° around a person (see Figure 6-3). In total, the dataset has 26500 videos of more than 200 frames each. 118 subjects perform 40 actions while always facing the front camera. A second camera is placed in one of the 7 other views, meaning that the temporal synchronization is only provided between front view and another view, but not between any views. This dataset allows studying actions from unusual views such as behind the person. We use the standard protocol Cross View II [Ji et al., 2018] which reports average accuracy of two splits: training with FV, V2, V4, V6 and testing with V1, V3, V5, V7, and vice versa.

6.4.2 Ablation study

We start with the cross-view-subject protocol on NTU. We first gradually increase the number of views in the training and test the models on all views. Baselines section of Table 6.1 summarizes these results. If we train with only 0°, i.e., $\mathcal{L}_a(v_1)$, the performance is high (81.7%) when tested on v_1, but significantly drops (40.2%) when tested on v_3. Adding more views in the training naturally reduces this gap, resulting in 59.9% and 73.0% accuracies when v_2 and v_3 are added to the training, respectively. However, adding naively more views does not increase the test accuracy of v_1 (81.7% versus 81.9%).

In order to study the effect of similarity training, we set $\mathcal{L}_a(v_1) + \mathcal{L}_a(v_2)$ as our baseline, i.e., $0°, 45°$ videos available at training. For this setup, we investigate two variants of the similarity training: (i) view-invariant features as explained in Section 6.3 and (ii) view-adapted features which will be explained next. For each variant, we also explore two ways of sampling the pairs: (i) only positive sampling ($\mathcal{L}_s$) and (ii) both positive and negative sampling ($\mathcal{L}_c$).

Invariance versus adaptation. We study a variant of our architecture which we refer to as adaptation. As illustrated in Figure 6-4(c-d), we define a mapping k which adapts the features h_2 to h_2' such that h_2' becomes closer to h_1. Specifically, we design k as a residual block of 2 fully connected layers with a ReLU nonlinearity in between, similar to [Cao et al., 2018; Rad et al., 2018]. The adaptation function k aims at learning the residual to transform feature representations from v_2 to v_1. The final action prediction of v_2 therefore becomes $f \circ k \circ g(x_{v_2})$. This architecture is particularly useful if updating the parameters of g is not desired. Then, one can freeze its parameters and only train for k.

The bottom part of the Table 6.1 shows the results obtained with these view-adapted features. First, we note that by freezing g_θ, we can already obtain 10% improvement (40.2% versus 51.0%). Jointly training further with the action losses and

Table 6.1 – Ablation on our cross-view-subject split of the NTU dataset. All networks are pre-trained on the first row, i.e., $\mathcal{L}_a(v_1)$. With abuse of notation, we denote training with v_1 videos as $\mathcal{L}_a(v_1)$. $\mathcal{L}_s$ and $\mathcal{L}_c$ denote similarity loss with only positive samples and both positive/negative samples, respectively. v_1, v_2, and v_3 correspond to 0°, 45°, and 90°. v' denotes adaptation (see Section 6.4.2).

Baselines	$v1$	$v1'$	$v2$	$v2'$	$v3$	$v3'$
$\mathcal{L}_a(v_1)$	81.7		63.3		40.2	
$\mathcal{L}_a(v_1) + \mathcal{L}_a(v_2)$	81.8		77.6		59.9	
$\mathcal{L}_a(v_1) + \mathcal{L}_a(v_2) + \mathcal{L}_a(v_3)$	81.9		79.5		73.0	
View-invariant features	$v1$		$v2$		$v3$	
$\mathcal{L}_s(v_1, v_2) + \mathcal{L}_a(v_1)$	82.8		74.1		55.2	
$\mathcal{L}_s(v_1, v_2) + \mathcal{L}_a(v_1) + \mathcal{L}_a(v_2)$	83.7		**80.6**		**64.4**	
$\mathcal{L}_c(v_1, v_2) + \mathcal{L}_a(v_1)$	**84.0**		73.3		52.3	
$\mathcal{L}_c(v_1, v_2) + \mathcal{L}_a(v_1) + \mathcal{L}_a(v_2)$	83.5		80.0		62.2	
View-adapted features	$v1$	$v1'$	$v2$	$v2'$	$v3$	$v3'$
$\mathcal{L}_s(v_1, v_2')$ [frozen g_β]	81.7	80.0	63.3	73.2	40.2	51.0
$\mathcal{L}_s(v_1, v_2') + \mathcal{L}_a(v_1)$	83.8	82.2	72.4	77.0	49.4	57.4
$\mathcal{L}_s(v_1, v_2') + \mathcal{L}_a(v_1) + \mathcal{L}_a(v_2')$	83.0	81.7	78.7	**79.4**	60.1	**62.6**
$\mathcal{L}_c(v_1, v_2')$ [frozen g_β]	81.7	79.0	63.3	70.2	40.2	49.1
$\mathcal{L}_c(v_1, v_2') + \mathcal{L}_a(v_1)$	**84.2**	82.7	70.8	75.9	46.6	55.7
$\mathcal{L}_c(v_1, v_2') + \mathcal{L}_a(v_1) + \mathcal{L}_a(v_2')$	83.9	82.7	77.3	**79.4**	56.3	62.4

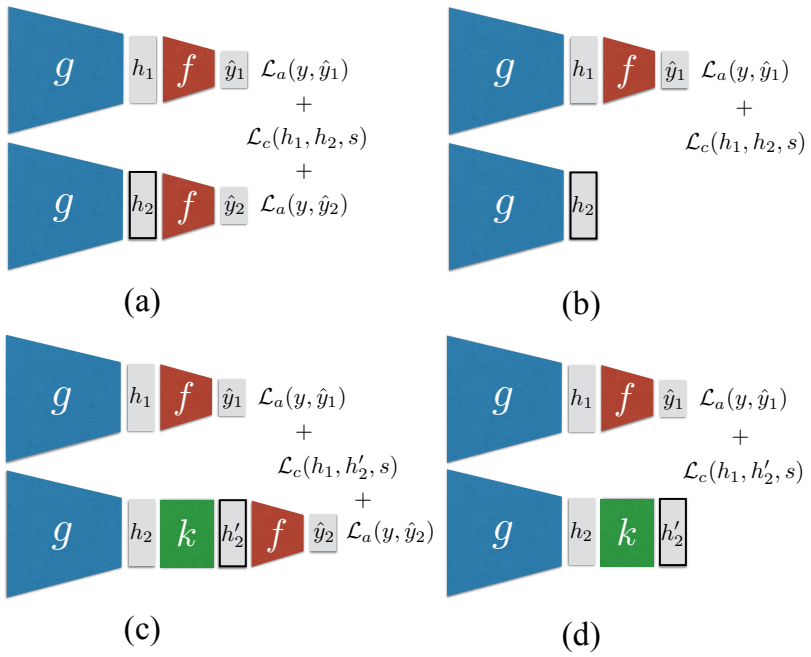


Figure 6-4 – Our model (a) and its variants (b-d) for which we perform ablation study. (c-d) describe the adaptation model with the function k that maps v_2 features to v_1. (b-d) remove the action supervision on the second view. The results are summarized in Table 6.1.

updating g results in 62.6% accuracy, outperforming the baseline 59.9%. Updating g parameters helps making the representation more suitable for adaptation.

Compared with adaptation, view-invariant features overall perform slightly better while being simpler to train. Finally, the view-invariant features outperform the baseline significantly (64.4% versus 59.9%). It is interesting to note that even the source view v_1 improves with both view-invariant and view-adapted feature training. The performance increases from 81.8% to 84.0% and to 84.2% for invariance and adaptation, respectively.

Action supervision on both views. Table 6.1 also checks whether having both action losses $\mathcal{L}_a(v_1)$ and $\mathcal{L}_a(v_2)$ is necessary. Figure 6-4(b-d) illustrates versions of our architecture without action supervision on v_2. In this case, videos of the second view only serve for the representation supervision. We find that it is important to

Table 6.2 – Results on the N-UCLA dataset using single-view at training. For the cross-dataset experiments, we define the action loss $\mathcal{L}_a$ on N-UCLA, and the similarity loss on NTU. $\mathcal{L}_c$(NTU) denotes contrastive loss on the NTU dataset where we use all pairs from the three views and sample both positives and negatives.

Single-view training	V^3	V^2	V^1
$\mathcal{L}_a(V_1)$	75.2	77.5	-
$\mathcal{L}_a(V_1) + \mathcal{L}_c$(NTU)	**81.7**	**78.2**	-
$\mathcal{L}_a(V_2)$	**82.6**	-	43.6
$\mathcal{L}_a(V_2) + \mathcal{L}_c$(NTU)	81.5	-	**62.6**
$\mathcal{L}_a(V_3)$	-	71.3	45.1
$\mathcal{L}_a(V_3) + \mathcal{L}_c$(NTU)	-	**71.9**	**58.8**

keep both action supervisions to achieve the best results (64.4% versus 55.2%).

Positive and negative sampling. In Section 6.3, we described the similarity loss based only on positive samples for the two-view setting. Here, we study whether a contrastive loss that also requires negative samples is needed. We therefore determine negative samples as videos from v_2 of the same subject performing a different action. Although negative samples are commonly adopted in metric learning [Sermanet et al., 2018; Sigurdsson et al., 2018], we do not observe improvements by using negatives. This can be explained by the presence of action loss that already separates the negatives. Table 6.1 compares sampling only positives ($\mathcal{L}_s$) with sampling both positives and negatives ($\mathcal{L}_c$). While the two performances are mostly similar, $\mathcal{L}_s$ alone performs slightly better and is simpler.

6.4.3 Cross-dataset training

Here, we explore the scenario of having videos from a single view with action labels. For this experiment, we use the medium-sized N-UCLA dataset. First, we train the baseline networks for each view in the dataset: $\mathcal{L}_a(V_1)$, $\mathcal{L}_a(V_2)$, and $\mathcal{L}_a(V_3)$. We test these networks on each of the other views. In Table 6.2, we observe that

Table 6.3 – Results on the N-UCLA dataset with the standard protocol with comparison to the state of the art. Column caption $V^k_{i,j}$ denotes training on views (i, j) and testing on view k. Both the action loss and the similarity loss are defined on N-UCLA.

Two-views training ($V_{i,j}$)	$V^3_{1,2}$	$V^2_{1,3}$	$V^1_{2,3}$	Avg
$\mathcal{L}_a(V_i) + \mathcal{L}_a(V_j)$	90.2	88.8	75.7	84.9
$\mathcal{L}_a(V_i) + \mathcal{L}_a(V_j) + \mathcal{L}_s(V_i, V_j)$	**91.1**	**89.0**	**80.5**	**86.9**
nCTE [Gupta et al., 2014]	68.6	68.3	52.1	63.0
NKTM [Rahmani and Mian, 2015]	75.8	73.3	59.1	69.4
TSN [Wang et al., 2016a] (by [Wang et al., 2018a])	84.5	80.6	76.8	80.6
ResNet50-3D [Baradel et al., 2018]	85.6	84.7	79.2	83.2
DA-net [Wang et al., 2018a]	86.5	82.7	83.1	84.2
Glimpse Clouds [Baradel et al., 2018]	90.1	89.5	83.4	87.6

Table 6.4 – Our model outperforms the state of the art on the recent UESTC dataset by using only RGB input.

Training		V1, V3, V5, V7					FV, V2, V4, V6					
Test		FV	V2	V4	V6	Avg_{even}	V1	V3	V5	V7	Avg_{odd}	Avg
VS-CNN [Ji et al., 2018]	Skeleton	87	54	71	60	68.0	87	58	60	87	73.0	70.5
JOULE [Hu et al., 2017] (by [Ji et al., 2018])	RGB	74	49	57	55	58.8	74	48	47	80	62.3	60.6
ResNeXt [Hara et al., 2018] (by [Ji et al., 2018])	RGB	51	40	54	39	46.0	52	44	48	52	49.0	47.5
Ours w/out similarity	RGB	**78.1**	63.1	76.6	46.4	66.1	**92.1**	80.9	85.5	86.1	86.2	76.1
Ours w/ similarity	RGB	76.7	**63.6**	**80.7**	**49.3**	**67.6**	91.9	**86.1**	**88.4**	**89.8**	**89.1**	**78.3**

Table 6.5 – We report on the standard protocols of NTU. Our method is on par with the state of the art while being complementary. Unlike [Baradel et al., 2018; Luo et al., 2018; Luvizon et al., 2018] we do not use modalities other than RGB during training. In contrast to [Wang et al., 2018a; Zolfaghari et al., 2017], our model does not input pre-computed modalities such as optical flow.

Method	Modality	CS	CV
[Shahroudy et al., 2016] Part-LSTM	Skeleton	62.9	70.3
[Liu et al., 2016] ST-LSTM	Skeleton	69.2	77.7
[Liu et al., 2017a] GCA-LSTM	Skeleton	74.4	82.8
[Ke et al., 2017] MTLN	Skeleton	79.6	84.8
[Liu et al., 2017b] View-invariant	Skeleton	80.0	87.2
[Baradel et al., 2017] Hands attention	RGB+Skeleton	84.8	90.6
[Liu and Yuan, 2018] Pose evolution	RGB+Depth	91.7	95.3
[Baradel et al., 2017] Hands attention	RGB	75.6	80.5
[Liu and Yuan, 2018] Pose evolution	RGB	78.8	84.2
[Zolfaghari et al., 2017] Multi-stream	RGB	80.8	-
[Luvizon et al., 2018] Multi-task	RGB	85.5	-
[Baradel et al., 2018] Glimpse clouds	RGB	86.6	93.2
[Wang et al., 2018a] DA-Net	RGB	88.1	92.0
[Luo et al., 2018] Graph distillation	RGB	89.5	-
Ours w/out similarity	RGB	86.3	**90.8**
Ours w/ similarity	RGB	**87.1**	90.2

the performance is particularly low when networks are tested on V_1, indicating the difficulty of this split.

As detailed in Section 6.3, we supervise the similarity on a separate dataset. In this experiment, we use all three views from NTU to train the similarity. In practice, we randomly sample from the two datasets with equal probabilities. We do forward pass for all videos and define the loss only on actions for N-UCLA and only on similarity for NTU samples.

Table 6.2 shows the results of this cross-dataset experiment for each view. We observe significant improvements especially for the challenging V_1 test set. From the baseline cross-view test results, we can also deduce that V_1, V_2 and V_2, V_3 are similar pairs, e.g., when tested on V_2, training with V_1 gives better performance than training with V_3 (77.5% versus 71.3%). This can be the reason why similarity training helps the most on V_1, V_3 pair, i.e. 75.2% versus 81.7% and 45.1% versus 58.8%. We conclude that in the case of a challenging cross-view scenario, where the source view is drastically different than the target view, the similarity training, although from a different data source, significantly improves the generalization to the unseen viewpoint.

6.4.4 Comparison with the state of the art

In the following, we employ the standard protocols for the N-UCLA, UESTC, NTU datasets used in the experiments and compare our results to other works.

Table 6.3 shows the results of training on two views on the N-UCLA dataset. First, we note the relatively lower improvement obtained with the similarity training (86.9% versus 84.9%) compared to using single-view. This can be explained by the unclear separation of viewpoints in the dataset and the fact that the videos are not perfectly synchronized. Especially V_1 videos are much longer than V_2-V_3 videos and V_2-V_3 videos are temporally better aligned. Thus, for the $V^1_{2,3}$ split, we obtain better improvement with similarity (from 75.7% to 80.5%), than for $V^3_{1,2}$ and $V^2_{1,3}$. Next, we

Figure 6-5 – Qualitative results are shown on NTU for 0° and 90° test views. The first line below each image is the prediction of the baseline model trained on 0° and 45° training views. The second line is the prediction of our similarity model. Top row shows our improvement over the baseline. The bottom row illustrates failure modes. Fine-grained categories cause confusions for both models.

compare to the state of the art on this dataset. Our results are on par with Baradel et al. [2018] which uses additional human pose supervision. The view-specific features of DA-Net [Wang et al., 2018a] is below our performance although they use both RGB and optical flow inputs.

We further report the performance of our method on the recently released UESTC dataset. Table 6.4 compares our results to the baselines reported by [Ji et al., 2018]. We outperform the RGB-based methods ResNeXt [Hara et al., 2018] and JOULE [Hu et al., 2017] by a large margin. Note that the lowest performances among the individual test views are V2 and V6, which correspond to ±90° side views. This confirms the large appearance changes for side-view actions noted earlier on the NTU dataset. Training with even views, and testing with odd views is better than the opposite mainly because the dataset has more training samples in this split. Similarity improves over the baseline on the views further from the front view.

In Table 6.5, we compare our results to the state-of-the-art methods on standard NTU splits. For the cross-subject setting (CS), all views are seen at training and at test. We define the similarity between any synchronous training pairs, e.g., 0-

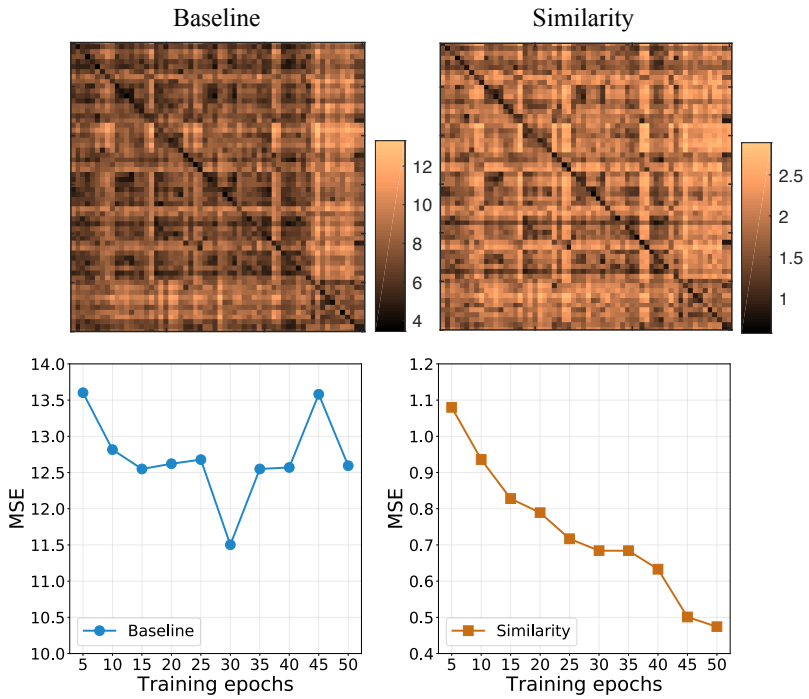


Figure 6-6 – We compare our model with the baseline by measuring the similarity between cross-view videos on NTU. Top: Matrices measure $\mathcal{L}_s$ between 60 synchronous test videos of v_1 and v_3, one per category. Note the different range between the two matrices. Bottom: The evolution of similarity during training.

45°, 0-90°, 45-90°. For the cross-view setting (CS), we define the similarity between the training views, 0-90°, and test with the 45° view. Our results on both splits achieve state-of-the-art performance only with RGB data. However, similarity does not improve on the CV split. This can be explained by the same subjects appearing both in training and test with only 45° difference. In comparison, [Baradel et al., 2018; Luvizon et al., 2018] both use pose information during training. [Luo et al., 2018] uses other modalities such as depth, skeleton and optical flow during training. Similarly, [Wang et al., 2018a] uses optical flow as input and employs a two-stream architecture. Moreover, [Zolfaghari et al., 2017] uses three-stream inputs including pre-computed optical flow and segmentation. Our method is complementary and can be integrated to these approaches.

6.4.5 Qualitative analysis

We next analyze our similarity model with comparison to the baseline qualitatively. For this analysis, we use the models trained on NTU cross-subject-view protocol with v_1 and v_2 training videos. We test both models on the test videos from views v_1 and v_3.

Figure 6-6 plots our similarity measure $\mathcal{L}_s$ for the two models. The top matrices compute the similarity between 60 temporally synchronous test videos from v_1 and v_3, one pair per action category. Our model encourages similarity on the diagonal both with the action supervision and the similarity supervision. This results in a higher contrast with respect to the off-diagonal elements when compared with the baseline. Note that the range for the similarity values is different for the two models and is much higher for the baseline. We further denote the grouping between the first 50 action classes. In NTU, the last 10 categories are two-person interactions, which make them visually different and easier to separate from the rest.

The bottom row of Figure 6-6 shows the evolution of similarity during training on validation videos from v_1 and v_2. As expected, the $\mathcal{L}_s$ measure remains high and

fluctuates in the baseline model because there are no constraints on the features. On the other hand, our model minimizes the similarity loss simultaneously with the action losses.

We further show qualitative results on test videos of v_1 and v_3 in Figure 6-5. Every image is accompanied with two predicted categories: the first one with the baseline model, and the second with ours. Top row illustrates cases where our model improves the v_3 predictions. Bottom row shows failure modes which are often common between the two models. Failures are mainly due to the fine-grained classes such as 'take off a shoe' and 'wear a shoe'. In the leftmost example, the 'wipe face' category is confused with 'touch head' from the front view, and with the 'sneeze/cough' category from the side. Ambiguous visual cues appearing in different views remain challenging.

6.5 Conclusions

We explored view-invariant video representations for action recognition. We proposed a conceptually simple, but effective supervision that can be integrated into CNN training for several scenarios to improve action recognition from unseen viewpoints. Our method achieves state-of-the-art results on multiple benchmarks given only RGB data. Future work could exploit synthetic datasets, as in Chapter 3, to scale up the temporally synchronous multi-view data. Another future direction could be to use a similarity definition parameterized by 3D human pose predictions.

Chapter 7

Discussion

In this chapter, we conclude this thesis by providing a summary of its contributions (Section 7.1) and outlining directions for future work (Section 7.2).

7.1 Summary of contributions

This thesis has addressed two areas of human understanding. For human body analysis, our contributions are twofold:

- In Chapter 3, we have generated a large-scale synthetic dataset of people with many ground truth modalities, which allowed us to train CNNs for human body part segmentation and depth estimation tasks, for which acquiring labels is expensive. We investigated the generalization capabilities of models trained on synthetic data and studied the effect of different components in synthetic data generation.

- In Chapter 4, we have introduced our BodyNet architecture and showed the advantages of volumetric representation for body shape estimation. Our experiments have demonstrated the benefits of multi-task training of intermediate tasks such as 2D/3D pose estimation and 2D part segmentation. We found it critical to

enforce a re-projection loss on the volumetric output to obtain confident predictions at extremities of the body. Finally, we extend our framework for 3D body part segmentation and for capturing 3D cloth deformations.

For human action recognition, our contributions have been the following:

- In Chapter 5, we have presented an extensive study on the benefits of long-term temporal convolutions for learning discriminative action representations. We used 3D convolutional neural networks on videos with high temporal resolution. We followed a two-stream approach where we combined long-term RGB video and long-term optical flow video inputs to 3D CNNs. We further showed that the quality of the optical flow algorithm has a significant influence on the performance of action recognition.

- In Chapter 6, we have leveraged multi-view videos for learning view-independent action representations. We formulated a similarity training where two temporally synchronous videos that capture the same action should have the maximum similarity in their embeddings. Our experiments showed that such framework improves the generalization to unseen viewpoints. We further addressed the case of a single view available with action labels. We performed a joint training for action classification on single-view videos and similarity on other multi-view videos. This mixed training scheme obtained significant improvements.

7.2 Future work

Synthetic-real domain gap. We expect additional improvements could originate from more appropriate synthetic data. However, currently, the optimal recipe for creating synthetic datasets is unknown. A detailed study on different aspects of synthetic data generation would indicate which direction to explore. For example, we have no evidence that photo-realism is necessary for successful generalization. In our

experiments in Chapter 3, we showed that the diversity is important. However, how to capture the real distributions for both pixel and target statistics is a challenging future direction.

Another potential extension would explore different domain adaptation techniques. The purpose of our work in Chapter 3 was to show that the SURREAL dataset sufficiently generalizes without the need for complex domain adaptation. Nevertheless, to obtain competitive performance on challenging images, we need to make use of real training data.

Body shape from in-the-wild images. The current methods for body shape estimation are mostly limited to constrained settings. The approach we presented in Chapter 4 is not robust against occlusions or extreme poses. Furthermore, the body is approximately centered in the input image. To address these issues, we could integrate a person detector in the pipeline. Occlusions can be artificially generated during training. In this case, a special care should be taken to adapt the re-projection loss. One other promising direction is to leverage unlabeled images in weakly or self-supervised settings.

Synthetic data for action recognition. Part I of this thesis uses synthetic data for body analysis. Synthetic data is a useful training resource for low-level tasks such as body shape estimation. It is unknown whether it can also be used for augmenting action recognition datasets. However, action recognition is relatively high-level task which requires reasoning of interactions with other objects and environments. We believe that this is difficult to simulate. Instead, we are planning to investigate the multi-view aspect of synthetic videos especially for extending the work presented in Chapter 6. View-independent representations can be learned by enforcing similarity between human motions rendered from different viewpoints.

Closing the loop: pose & action. We have presented work on the low-level human body analysis (Part I: Chapters 3, 4) and human action recognition (Part II: Chapters 5, 6). Despite close relations between these two domains, there is little interaction between them in the concurrent work. Intuitively, the knowledge of body pose should help the recognition of actions. Vice versa, the knowledge of action should constrain the possible body configurations. Despite recent efforts on integrating pose estimation and action recognition in joint frameworks, we have still not witnessed significant gains from such a joint approach. We believe that this is an interesting next step, especially given the success of current pose estimation methods.

Annex A

Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding

Computer vision has a great potential to help our daily lives by searching for lost keys, watering flowers or reminding us to take a pill. To succeed with such tasks, computer vision methods need to be trained from real and diverse examples of our daily dynamic scenes. While most of such scenes are not particularly exciting, they typically do not appear on YouTube, in movies or TV broadcasts. So how do we collect sufficiently many diverse but *boring* samples representing our lives? We propose a novel Hollywood in Homes approach to collect such data. Instead of shooting videos in the lab, we ensure diversity by distributing and crowdsourcing the whole process of video creation from script writing to video recording and annotation. Following this procedure we collect a new dataset, *Charades*, with hundreds of people recording videos in their own homes, acting out casual everyday activities. The dataset is composed of 9,848 annotated videos with an average length of 30 seconds, showing activities of 267 people from three continents. Each video is annotated by multiple free-text descriptions, action labels, action intervals and classes of interacted objects. In total,

Charades provides 27,847 video descriptions, 66,500 temporally localized intervals for 157 action classes and 41,104 labels for 46 object classes. Using this rich data, we evaluate and provide baseline results for several tasks including action recognition and automatic description generation. We believe that the realism, diversity, and casual nature of this dataset will present unique challenges and new opportunities for computer vision community.

A.1 Introduction

Large scale visual learning fueled by huge datasets has changed the computer vision landscape [Deng et al., 2009; Zhou et al., 2014]. Given the source of this data, it's not surprising that most of our current success is biased towards static scenes and objects in Internet images. As we move forward into the era of AI and robotics, however, new questions arise. How do we learn about different states of objects (e.g., cut vs. whole)? How do common activities affect changes of object states? In fact, it is not even yet clear if the success of the Internet pre-trained recognition models will transfer to real-world settings where robots equipped with our computer vision models should operate.

Shifting the bias from Internet images to real scenes will most likely require collection of new large-scale datasets representing activities of our boring everyday life: getting up, getting dressed, putting groceries in fridge, cutting vegetables and so on. Such datasets will allow us to develop new representations and to learn models with the right biases. But more importantly, such datasets representing people interacting with objects and performing natural action sequences in typical environments will finally allow us to learn common sense and contextual knowledge necessary for high-level reasoning and modeling.

But how do we find these boring videos of our daily lives? If we search common activities such as "drinking from a cup", "riding a bike" on video sharing websites such as YouTube, we observe a highly-biased sample of results (see Figure A-1). These results are biased towards entertainment—boring videos have no viewership and hence no reason to be uploaded on YouTube!

In this work, we propose a novel *Hollywood in Homes* approach to collect a large-scale dataset of boring videos of daily activities. Standard approaches in the past have used videos downloaded from the Internet [Caba Heilbron et al., 2015; Liu et al., 2009; Gorban et al., 2015; Karpathy et al., 2014; Kuehne et al., 2011; Soomro et al., 2012] gathered from movies [Laptev et al., 2008; Rodriguez et al., 2008; Rohrbach et al.,

Figure A-1 – Comparison of actions in the Charades dataset and on YouTube: *Reading a book*, *Opening a refrigerator*, *Drinking from a cup*. YouTube returns entertaining and often atypical videos, while *Charades* contains typical everyday videos.

2015] or recorded in controlled environments [Schüldt et al., 2004; Gorelick et al., 2007; Rohrbach et al., 2012; Oh et al., 2011; Kuehne et al., 2014; Rohrbach et al., 2014]. Instead, as the name suggests: we take the Hollywood filming process to the homes of hundreds of people on Amazon Mechanical Turk (AMT). AMT workers follow the three steps of filming process: (1) script generation; (2) video direction and acting based on scripts; and (3) video verification to create one of the largest and most diverse video dataset of daily activities.

There are threefold advantages of using the *Hollywood in Homes* approach for dataset collection: (a) Unlike datasets shot in controlled environments (e.g., MPII [Rohrbach et al., 2012]), crowdsourcing brings in diversity which is essential for generalization. In fact, our approach even allows the same script to be enacted by multiple people; (b) crowdsourcing the script writing enhances the coverage in terms of scenarios and reduces the bias introduced by generating scripts in labs; and (c) most importantly, unlike for web videos, this approach allows us to control the composition and the length of video scenes by proposing the vocabulary of scenes, objects and actions during script generation.

The Charades v1.0 Dataset

Charades is our large-scale dataset with a focus on common household activities collected using the Hollywood in Homes approach. The name comes from of a popular American word guessing game where one player acts out a phrase and the other players guess what phrase it is. In a similar spirit, we recruited hundreds of people from Amazon Mechanical Turk to act out a paragraph that we presented to them. The workers additionally provide action classification, localization, and video description annotations. The first publicly released version of our *Charades* dataset will contain 9,848 videos of daily activities 30.1 seconds long on average (7,985 training and 1,863 test). The dataset is collected in 15 types of indoor scenes, involves interactions with 46 object classes and has a vocabulary of 30 verbs leading to 157 action classes. It has 66,500 temporally localized actions, 12.8 seconds long on average, recorded by 267 people in three continents, and over 15% of the videos have more than one person. We believe this dataset will provide a crucial stepping stone in developing action representations, learning object states, human object interactions, modeling context, object detection in videos, video captioning and many more. The dataset is publicly available in the project page [Charades project page].

Contributions The contributions of our work are three-fold: (1) We introduce the Hollywood in Homes approach to data collection, (2) we collect and release the first crowdsourced large-scale dataset of boring household activities, and (3) we provide extensive baseline evaluations.

The KTH action dataset [Schüldt et al., 2004] paved the way for algorithms that recognized human actions. However, the dataset was limited in terms of number of categories and enacted in the same background. In order to scale up the learning and the complexity of the data, recent approaches have instead tried collecting video datasets by downloading videos from Internet. Therefore, datasets such as UCF101 [Soomro et al., 2012], Sports1M [Karpathy et al., 2014] and others [Kuehne et al., 2011; Liu et al., 2009; Gorban et al., 2015] appeared and presented more chal-

lenges including background clutter, and scale. However, since it is impossible to find boring daily activities on Internet, the vocabulary of actions became biased towards more sports-like actions which are easy to find and download.

There have been several efforts in order to remove the bias towards sporting actions. One such commendable effort is to use movies as the source of data [Marszałek et al., 2009; Ferrari et al., 2009]. Recent papers have also used movies to focus on the video description problem leading to several datasets such as MSVD[Chen and Dolan, 2011], M-VAD [Torabi et al., 2015], and MPII-MD [Rohrbach et al., 2015]. Movies however are still exciting (and a source of entertainment) and do not capture the scenes, objects or actions of daily living. Other efforts have been to collect in-house datasets for capturing human-object interactions [Gupta and Davis, 2007] or human-human interactions [Ryoo and Aggarwal, 2009]. Some relevant big-scale efforts in this direction include MPII Cooking [Rohrbach et al., 2012], TUM Breakfast [Kuehne et al., 2014], and the TACoS Multi-Level [Rohrbach et al., 2014] datasets. These datasets focus on a narrow domain by collecting the data in-house with a fixed background, and therefore focus back on the activities themselves. This allows for careful control of the data distribution, but has limitations in terms of generalizability, and scalability. In contrast, PhotoCity [Tuite et al., 2011] used the crowd to take pictures of landmarks, suggesting that the same could be done for other content at scale.

Another relevant effort in collection of data corresponding to daily activities and objects is in the domain of ego-centric cameras. For example, the Activities of Daily Living dataset [Pirsiavash and Ramanan, 2012] recorded 20 people performing unscripted, everyday activities in their homes in first person, and another extended that idea to animals [Iwashita et al., 2014]. These datasets provide a challenging task but fail to provide diversity which is crucial for generalizability. It should however be noted that these kinds of datasets could be crowdsourced similarly to our work.

The most related dataset is the recently released ActivityNet dataset [Caba Heil-

Table A.1 – Comparison of Charades with other video datasets.

	Actions per video	Classes	Labelled instances	Total videos	Origin	Type	Temporal localization
Charades v1.0	6.8	157	67K	10K	267 Homes	Daily Activities	Yes
ActivityNet [Caba Heilbron et al., 2015]	1.4	203	39K	28K	YouTube	Human Activities	Yes
UCF101 [Soomro et al., 2012]	1	101	13K	13K	YouTube	Sports	No
HMDB51 [Kuehne et al., 2011]	1	51	7K	7K	YouTube/Movies	Movies	No
THUMOS'15 [Gorban et al., 2015]	1-2	101	21K+	24K	YouTube	Sports	Yes
Sports 1M [Karpathy et al., 2014]	1	487	1.1M	1.1M	YouTube	Sports	No
MPII-Cooking [Rohrbach et al., 2012]	46	78	13K	273	30 In-house actors	Cooking	Yes
ADL [Pirsiavash and Ramanan, 2012]	22	32	436	20	20 Volunteers	Ego-centric	Yes
MPII-MD [Rohrbach et al., 2015]	Captions	Captions	68K	94	Movies	Movies	No

bron et al., 2015]. It includes actions of daily living downloaded from YouTube. We believe the ActivityNet effort is complementary to ours since their dataset is uncontrolled, slightly biased towards non-boring actions and biased in the way the videos are professionally edited. On the other hand, our approach focuses more on action sequences (generated from scripts) involving interactions with objects. Our dataset, while diverse, is controlled in terms of vocabulary of objects and actions being used to generate scripts. In terms of the approach, Hollywood in Homes is also related to [Zitnick and Parikh, 2013]. However, [Zitnick and Parikh, 2013] only generates synthetic data. A comparison with other video datasets is presented in Table A.1. To the best of our knowledge, our approach is the first to demonstrate that workers can be used to collect a vision dataset by filming themselves at such a large scale.

A.2 Hollywood in Homes

We now describe the approach and the process involved in a large-scale video collection effort via AMT. Similar to filming, we have a three-step process for generating a video. The first step is generating the script of the indoor video. The key here is to allow workers to generate diverse scripts yet ensure that we have enough data for each category. The second step in the process is to use the script and ask workers to record a video of that sentence being acted out. In the final step, we ask the workers to verify if the recorded video corresponds to script, followed by an annotation

procedure.

A.2.1 Generating Scripts

In this work we focus on indoor scenes, hence, we group together rooms in residential homes (*Living Room*, *Home Office*, etc.). We found 15 types of rooms to cover most of typical homes, these rooms form the scenes in the dataset. In order to generate the *scripts* (a text given to workers to act out in a video), we use a vocabulary of objects and actions to guide the process. To understand what objects and actions to include in this vocabulary, we analyzed 549 movie scripts from popular movies in the past few decades. Using both term-frequency (TF) and TF-IDF [Salton and McGill, 1986] we analyzed which nouns and verbs occur in those rooms in these movies. From those we curated a list of 40 objects and 30 actions to be used as seeds for script generation, where objects and actions were chosen to be generic for different scenes.

To harness the creativity of people, and understand their bias towards activities, we crowdsourced the script generation as follows. In the AMT interface, a single scene, 5 randomly selected objects, and 5 randomly selected actions were presented to workers. Workers were asked to use two objects and two actions to compose a short paragraph about activities of one or two people performing realistic and commonplace activities in their home. We found this to be a good compromise between controlling what kind of words were used and allowing the users to impose their own human bias on the generation. Some examples of generated scripts are shown in Figure A-2. (see the website for more examples). The distribution of the words in the dataset is presented in Figure A-3.

A.2.2 Generating Videos

Once we have scripts, our next step is to collect videos. To maximize the diversity of scenes, objects, clothing and behaviour of people, we ask the workers themselves to record the 30 second videos by following collected scripts.

AMT is a place where people commonly do quick tasks in the convenience of their homes or during downtime at their work. AMT has been used for annotation and editing but can we do content creation via AMT? During a pilot study we asked workers to record the videos, and until we paid up to $3 per video, no worker picked up our task. (For comparison, to annotate a video [Sigurdsson et al., 2016a]: 3 workers × 157 questions × 1 second per question × $8/h salary = $1.) To reduce the base cost to a more manageable $1 per video, we have used the following strategies:

Worker Recruitment. To overcome the inconvenience threshold, worker recruitment was increased through sign-up bonuses (211% increased new worker rate) where we awarded a $5 bonus for the first submission. This increased the total cost by 17%. In addition, "recruit a friend" bonuses ($5 if a friend submits 15 videos) were introduced, and were claimed by 4% of the workforce, generating indeterminate outreach to the community. US, Canada, UK, and, for a time, India were included in this study. The first three accounted for estimated 73% of the videos, and 59% of the peak collection rate.

Worker Retention. Worker retention was mitigated through performance bonuses every 15th video, and while only accounting for a 33% increase in base cost, significantly increased retention (34% increase in come-back workers), and performance (109% increase in output per worker).

Each submission in this phase was manually verified by other workers to enforce quality control, where a worker was required to select the corresponding sentence from a line-up after watching the video. The rate of collection peaked at 1225 per day from 72 workers. The final cost distribution was: 65% base cost per video, 21% performance bonuses, 11% recruitment bonuses, and 3% verification. The code and interfaces will be made publicly available along with the dataset.

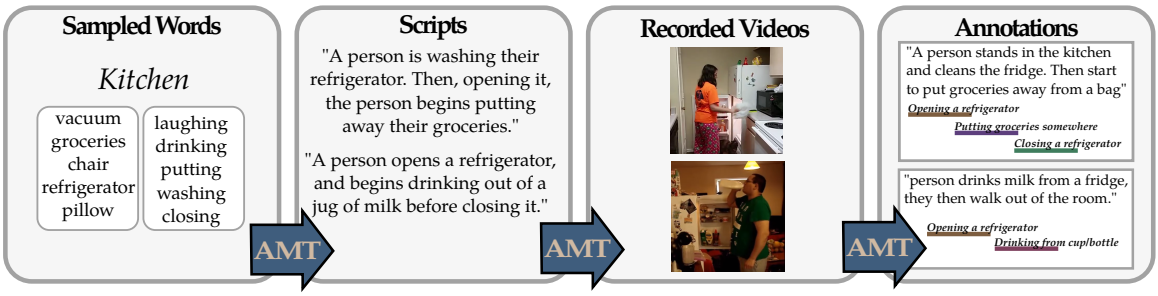


Figure A-2 – An overview of the three Amazon Mechanical Turk (AMT) crowdsourcing stages in the *Hollywood in Homes* approach.

A.2.3 Annotations

Using the generated scripts, all (verb,proposition,noun) triplets were analyzed, and the most frequent grouped into 157 action classes (e.g., *pouring into cup*, *running*, *folding towel*, etc.). The distribution of those is presented in Figure A-3.

For each recorded video we have asked other workers to watch the video and describe what they have observed with a sentence (this will be referred to as a *description* in contrast to the previous *script* used to generate the video). We use the original script and video descriptions to automatically generate a list of interacted objects for each video. Such lists were verified by the workers. Given the list of (verified) objects, for each video we have made a short list of 4-5 actions (out of 157) involving corresponding object interactions and asked the workers to verify the presence of these actions in the video.

In addition, to minimize the missing labels, we expanded the annotations by exhaustively annotating all actions in the video using state-of-the-art crowdsourcing practices Sigurdsson et al. [2016a], where we focused particularly on the test set.

Finally, for all the chosen action classes in each video, another set of workers was asked to label the starting and ending point of the activity in the video, resulting in a temporal interval of each action. A visualization of the data collection process is illustrated in Figure A-2. On the website we show numerous additional examples from the dataset with annotated action classes.

A.3 Charades v1.0 Analysis

Charades is built up by combining 40 objects and 30 actions in 15 scenes. This relatively small vocabulary, combined with open-ended writing, creates a dataset that has substantial coverage of a useful domain. Furthermore, these combinations naturally form action classes that allow for standard benchmarking. In Figure A-3 the distributions of action classes, and most common nouns/verbs/scenes in the dataset are presented. The natural world generally follows a long-tailed distribution [Zipf, 1935; Simoncelli and Olshausen, 2001], but we can see that the distribution of words in the dataset is relatively even. In Figure A-3 we also present a visualization of what scenes, objects, and actions occur together. By embedding the words based on their co-occurance with other words using T-SNE [Van der Maaten and Hinton, 2008], we can get an idea of what words group together in the videos of the dataset, and it is clear that the dataset possesses real-world intuition. For example, *food*, and *cooking* are close to *Kitchen*, but note that except for *Kitchen*, *Home Office*, and *Bathroom*, the scene is not highly discriminative of the action, which reflects common daily activities.

Since we have control over the data acquisition process, instead of using Internet search, there are on average 6.8 relevant actions in each video. We hope that this may inspire new and interesting algorithms that try to capture this kind of context in the domain of action recognition. Some of the most common pairs of actions measured in terms of normalized pointwise mutual information (NPMI), are also presented in Figure A-3. These actions occur in various orders and context, similar to our daily lives. For example, in Figure A-4 we can see that among these five videos, there are multiple actions occurring, and some are in common. We further explore this in Figure A-5, where for a few actions, we visualize the most probable actions to precede, and most probable actions to follow that action. As the scripts for the videos are generated by people imagining a boring realistic scenario, we find that these statistics reflect human behaviour.

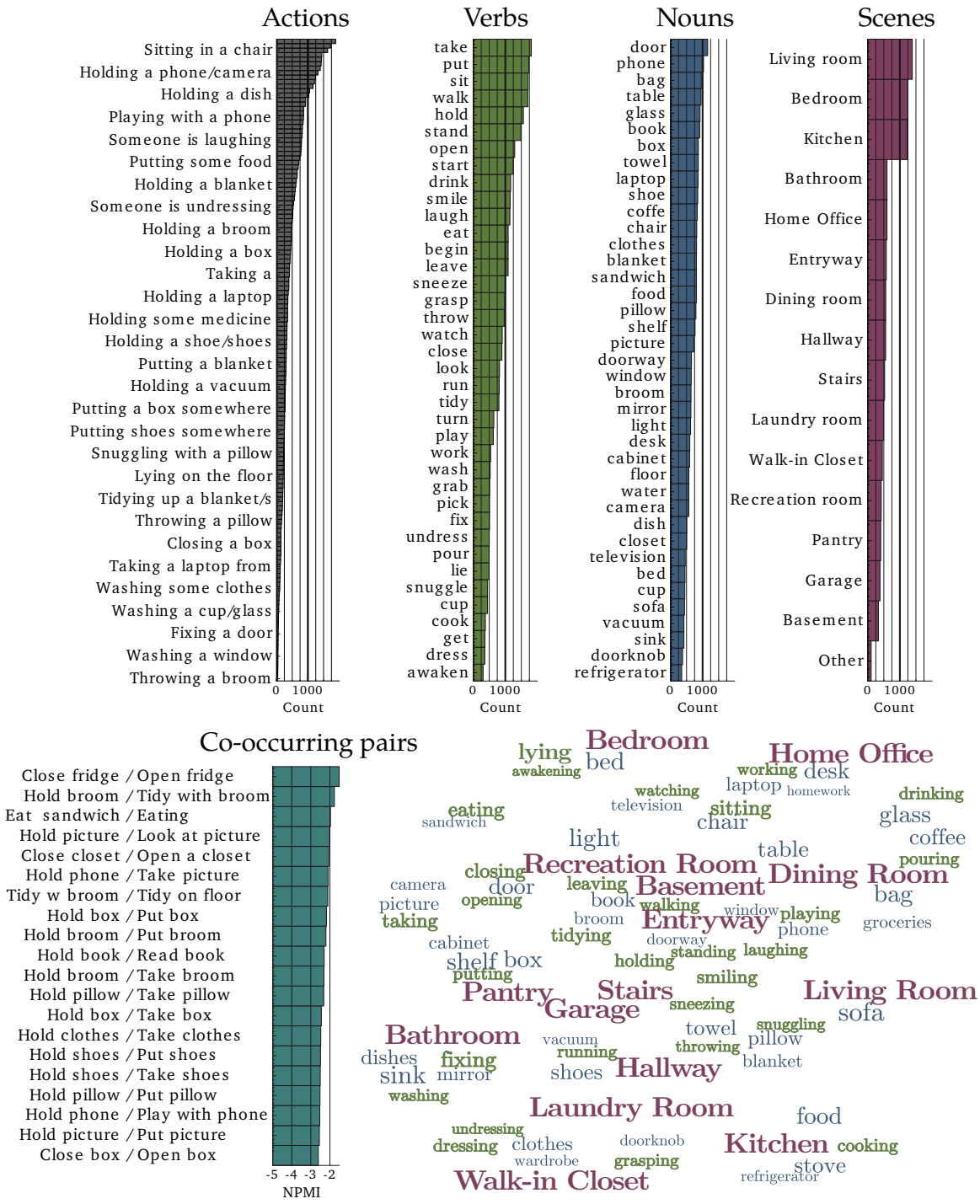


Figure A-3 – Statistics for actions (gray, every fifth label shown), verbs (green), nouns (blue), scenes (red), and most co-occurring pairs of actions (cyan). Co-occurrence is measured with normalized pointwise mutual information. In addition, a T-SNE embedding of the co-occurrence matrix is presented. We can see that while there are some words that strongly associate with each other (e.g., lying and bed), many of the objects and actions co-occur with many of the scenes. (Action names are abbreviated as necessary to fit space constraints.)

Figure A-4 – Keyframes from five videos in *Charades*. We see that actions occur together in many different configurations. (Shared actions are highlighed in color).

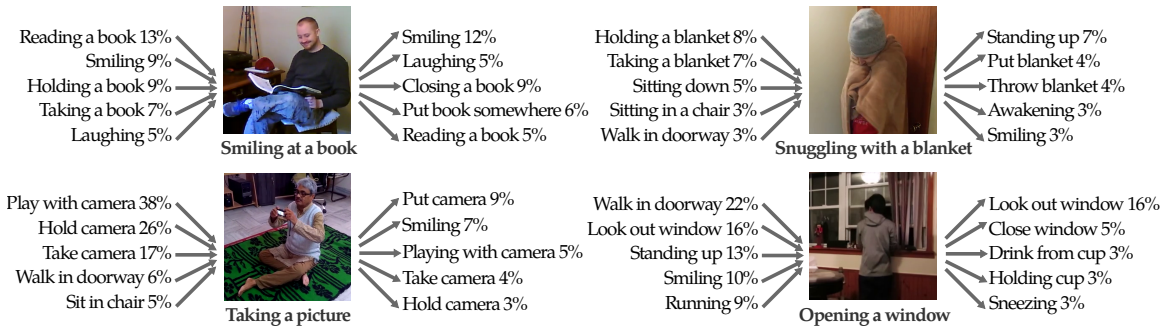


Figure A-5 – Selected actions from the dataset, along with the top five most probable actions before, and after the action. For example, when *Opening a window*, it is likely that someone was *Standing up* before that, and after opening, *Looking out the window*.

A.4 Applications

We run several state-of-the-art algorithms on Charades to provide the community with a benchmark for recognizing human activities in realistic home environments. Furthermore, the performance and failures of tested algorithms provide insights into the dataset and its properties.

Train/test set. For evaluating algorithms we split the dataset into train and test sets by considering several constraints: (a) the same worker should not appear in both training and test; (b) the distribution of categories over the test set should be similar to the one over the training set; (c) there should be at least 6 test videos and 25 training videos in each category; (d) the test set should not be dominated by a single worker. We randomly split the workers into two groups (80% in training) such that these constraints were satisfied. The resulting training and test sets contain 7,985 and 1,863 videos, respectively. The number of annotated action intervals are 49,809 and 16,691 for training and test.

A.4.1 Action Classification

Given a video, we would like to identify whether it contains one or several actions out of our 157 action classes. We evaluate the classification performance for several baseline methods. Action classification performance is evaluated with the standard mean average precision (mAP) measure. A single video is assigned to multiple classes and the distribution of classes over the test set is not uniform. The label precision for the data is 95.6%, measured using an additional verification step, as well as comparing against a ground truth made from 19 iterations of annotations on a subset of 50 videos. We now describe the baselines.

Improved trajectories. We compute improved dense trajectory features (IDT) [Wang and Schmid, 2013] capturing local shape and motion information with MBH, HOG and HOF video descriptors. We reduce the dimensionality of each descriptor

Table A.2 – mAP (%) for action classification with various baselines.

Random	C3D	AlexNet	Two-Stream-B	Two-Stream	IDT	Combined
5.9	10.9	11.3	11.9	14.3	17.2	18.6

Table A.3 – Action classification evaluation with the state-of-the-art approach on Charades. We study different parameters for improved trajectories, by reporting for different local descriptor sets and different number of GMM clusters. Overall performance improves by combining all descriptors and using a larger descriptor vocabulary.

	HOG	HOF	MBH	HOG+MBH	HOG+HOF+MBH
K=64	12.3	13.9	15.0	15.8	16.5
K=128	12.7	14.3	15.4	16.2	16.9
K=256	13.0	14.4	15.5	16.5	17.2

by half with PCA, and learn a separate feature vocabulary for each descriptor with GMMs of 256 components. Finally, we encode the distribution of local descriptors over the video with Fisher vectors [Perronnin et al., 2010]. A one-versus-rest linear SVM is used for classification. Training on untrimmed intervals gave the best performance.

Static CNN features. In order to utilize information about objects in the scene, we make use of deep neural networks pretrained on a large collection of object images. We experiment with VGG-16 [Simonyan and Zisserman, 2015] and AlexNet [Krizhevsky et al., 2012] to compute fc_6 features over 30 equidistant frames in the video. These features are averaged across frames, L2-normalized and classified with a one-versus-rest linear SVM. Training on untrimmed intervals gave the best performance.

Two-stream networks. We use the VGG-16 model architecture [Simonyan and Zisserman, 2015] for both networks and follow the training procedure introduced in [Simonyan and Zisserman, 2014], with small modifications. For the spatial network, we applied finetuning on ImageNet pre-trained networks with different dropout rates. The best performance was with 0.5 dropout rate and finetuning on all fully connected layers. The temporal network was first pre-trained on the UCF101 dataset and then similarly finetuned on conv4, conv5, and fc layers. Training on trimmed intervals

gave the best performance.

Balanced two-stream networks. We adapt the previous baseline to handle class imbalance. We balanced the number of training samples through sampling, and ensured each minibatch of 256 had at least 50 unique classes (each selected uniformly at random). Training on trimmed intervals gave the best performance.

C3D features. Following the recent approach from [Tran et al., 2015], we extract fc_6 features from a 3D convnet pretrained on the Sports-1M video dataset [Karpathy et al., 2014]. These features capture complex hierarchies of spatio-temporal patterns given an RGB clip of 16 frames. Similar to [Tran et al., 2015], we compute features on chunks of 16 frames by sliding 8 frames, average across chunks, and use a one-versus-rest linear SVM. Training on untrimmed intervals gave the best performance. Action classification results are presented in Table A.2, where we additionally consider *Combined* which combines all the other methods with late fusion.

Notably, the accuracy of the tested state-of-the-art baselines is much lower than in most currently available benchmarks. Consistently with several other datasets, IDT features [Wang and Schmid, 2013] outperform other methods by obtaining 17.2% mAP. To analyze these results, Figure A-6(left) illustrates the results for subsets of best and worst recognized action classes. We can see that while the mAP is low, there are certain classes that have reasonable performance, for example *Washing a window* has 62.1% AP. To understand the source of difference in performance for different classes, Figure A-6(right) illustrates AP for each action, sorted by the number of examples, together with names for the best performing classes. The number of actions in a class is primarily decided by the universality of the action (can it happen in any scene), and if it is common in typical households (writer bias). It is interesting to notice, that while there is a trend for actions with higher number of examples to have higher AP, it is not true in general, and actions such as *Sitting in chair*, and *Washing windows* have top-15 performance.

Delving even further, we investigate the confusion matrix for the *Combined* base-

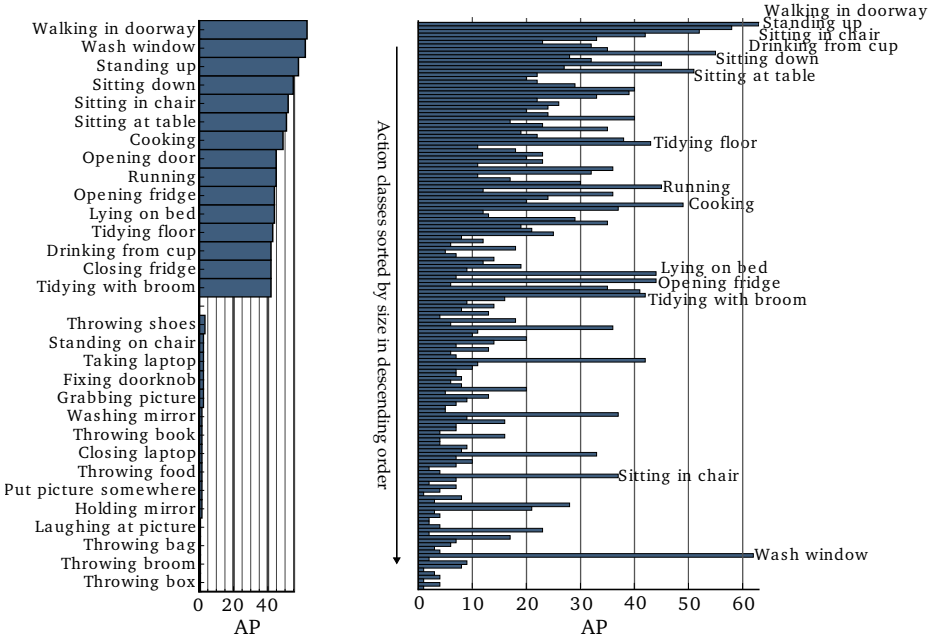


Figure A-6 – On the left classification accuracy for the 15 highest and lowest actions is presented for *Combined*. On the right, the classes are sorted by their size. The top actions on the left are annotated on the right. We can see that while there is a slight trend for smaller classes to have lower accuracy, many classes do not follow that trend.

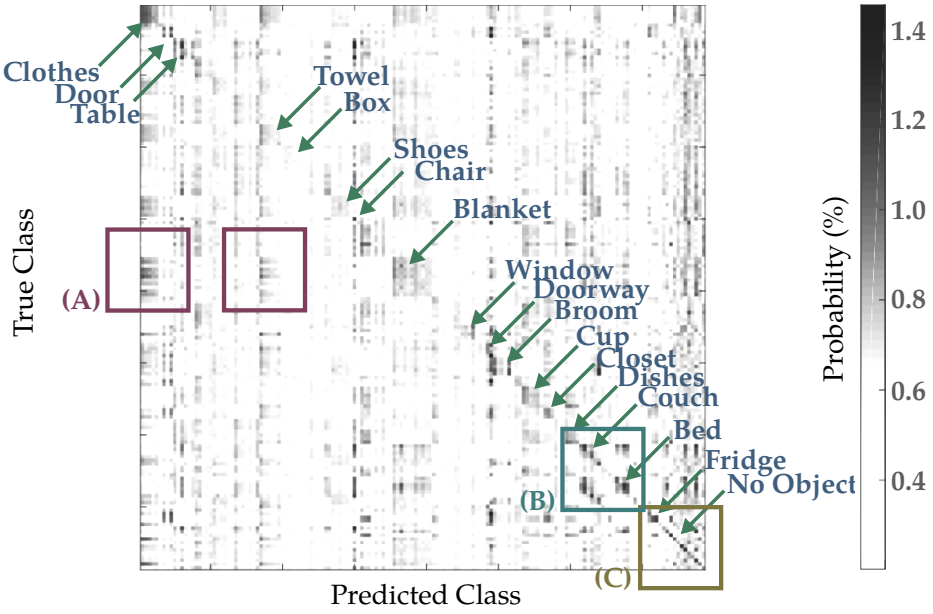


Figure A-7 – Confusion matrix for the *Combined* baseline on the classification task. Actions are grouped by the object being interacted with. Most of the confusion is with other actions involving the same object (s**q**uares on the diagonal), and we highlight some prominent objects. Note: (A) High confusion between actions using *Blanket*, *Clothes*, and *Towel*; (B) High confusion between actions using *Couch* and *Bed*; (C) Little confusion among actions with no specific object of interaction (e.g., *standing up*, *sneezing*).

line in Figure A-7, where we convert the predictor scores to probabilities and accumulate them for each class. For clearer analysis, the classes are sorted by the object being interacted with. The first aspect to notice is the squares on the diagonal, which imply that the majority of the confusion is among actions that interact with the same object (e.g., *Putting on clothes*, or *Taking clothes from somewhere*), and moreover, there is confusion among objects with similar functional properties. The most prominent squares are annotated with the object being shared among those actions. The figure caption contains additional observations. While there are some categories that show no clear trend, we can observe less confusion for many actions that have no specific object of interaction. Evaluation of action recognition on this subset results in 38.9% mAP, which is significantly higher than average. Recognition of fine-grained actions involving interactions with the same object class appears particularly difficult even for the best methods available today. We hope our dataset will encourage new methods addressing activity recognition for complex person-object interactions.

A.4.2 Sentence Prediction

Our final, and arguably most challenging task, concerns prediction of free-from sentences describing the video. Notably, our dataset contains sentences that have been used to create the video (*scripts*), as well as multiple video *descriptions* obtained manually for recorded videos. The scripts used to create videos are biased by the vocabulary, and due to the writer's imagination, generally describe different aspects of the video than descriptions. The description of the video by other people is generally simpler and to the point. Captions are evaluated using the CIDEr, BLEU, ROUGE, and METEOR metrics, as implemented in the COCO Caption Dataset [Chen et al., 2015b]. These metrics are common for comparing machine translations to ground truth, and have varying degrees of similarity with human judgement. For comparison, human performance is presented along with the baselines where workers were similarly asked to watch the video and describe what they observed. We now describe

Table A.4 – Sentence Prediction. In the *script* task one sentence is used as ground truth, and in the *description* task 2.4 sentences are used as ground truth on average. We find that S2VT is the strongest baseline.

	Script					***Description***				
	RW	Random	NN	S2VT	Human	RW	Random	NN	S2VT	Human
CIDEr	0.03	0.08	0.11	0.17	0.51	0.04	0.05	0.07	0.14	0.53
$BLEU_4$	0.00	0.03	0.03	0.06	0.10	0.00	0.04	0.05	0.11	0.20
$BLEU_3$	0.01	0.07	0.07	0.12	0.16	0.02	0.09	0.10	0.18	0.29
$BLEU_2$	0.09	0.15	0.15	0.21	0.27	0.09	0.20	0.21	0.30	0.43
$BLEU_1$	0.37	0.29	0.29	0.36	0.43	0.38	0.40	0.40	0.49	0.62
$ROUGE_L$	0.21	0.24	0.25	0.31	0.35	0.22	0.27	0.28	0.35	0.44
METEOR	0.10	0.11	0.12	0.13	0.20	0.11	0.13	0.14	0.16	0.24

the sentence prediction baselines in detail:

Random Words (RW): Random words from the training set.

Random Sentence (Random): Random sentence from the training set.

Nearest Neighbor (NN): Inspired by [Devlin et al., 2015] we simply use a 1-Nearest Neighbor baseline computed using AlexNet fc_7 outputs averaged over frames, and use the caption from that nearest neighbor in the training set.

S2VT: We use the S2VT method from [Venugopalan et al., 2015], which is a combination of a CNN, and a LSTM.

Table A.4 presents the performance of multiple baselines on the caption generation task. We both evaluate on predicting the *script*, as well as predicting the *description*. As expected, we can observe that descriptions made by people after watching the video are more similar to other descriptions, rather than the scripts used to generate the video. Table A.4 also provides insight into the different evaluation metrics, and it is clear that CIDEr offers the highest resolution, and most similarity with human judgement on this task. In Figure A-8 few examples are presented for the highest scoring baseline (S2VT). We can see that while the language model is accurate (the sentences are coherent), the model struggles with providing relevant captions, and tends to slightly overfit to frequent patterns in the data (e.g., *drinking from a glass/cup*).

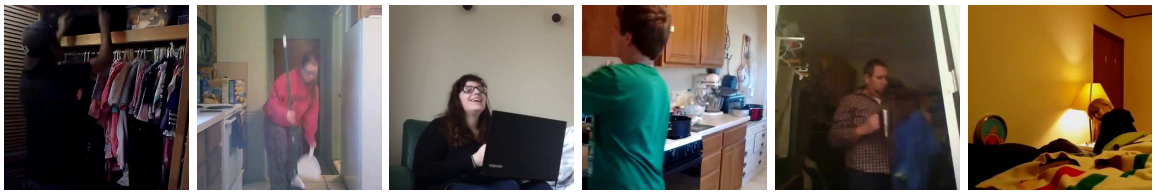


Figure A-8 – Three generated captions that scored low on the CIDEr metric (red), and three that scored high (green) from the strongest baseline (S2VT). We can see that while the captions are fairly coherent, the captions lack sufficient relevance.

A.5 Conclusions

We proposed a new approach for building datasets. Our Hollywood in Homes approach allows not only the labeling, but the data gathering process to be crowd-sourced. In addition, *Charades* offers a novel large-scale dataset with diversity and relevance to the real world. We hope that Charades and Hollywood in Homes will have the following benefits for our community:

(1) Training data: Charades provides a large-scale set of 66,500 annotations of actions with unique realism.

(2) A benchmark: Our publicly available dataset and provided baselines enable benchmarking future algorithms.

(3) Object-action interactions: The dataset contains significant and intricate object-action relationships which we hope will inspire the development of novel computer vision techniques targeting these settings.

(4) A framework to explore novel domains: We hope that many novel datasets in new domains can be collected using the Hollywood in Homes approach.

(5) Understanding daily activities: Charades provides data from a unique human-generated angle, and has unique attributes, such as complex co-occurrences of activities. This kind of realistic bias, may provide new insights that aid robots equipped with our computer vision models operating in the real world.

Annex B

Learning joint reconstruction of hands and manipulated objects

Estimating hand-object manipulations is essential for interpreting and imitating human actions. Previous work has made significant progress towards reconstruction of hand poses and object shapes in isolation. Yet, reconstructing hands and objects during manipulation is a more challenging task due to significant occlusions of both the hand and object. While presenting challenges, manipulations may also simplify the problem since the physics of contact restricts the space of valid hand-object configurations. For example, during manipulation, the hand and object should be in contact but not interpenetrate. In this work, we regularize the joint reconstruction of hands and objects with manipulation constraints. We present an end-to-end learnable model that exploits a novel contact loss that favors physically plausible hand-object constellations. Our approach improves grasp quality metrics over baselines, using RGB images as input. To train and evaluate the model, we also propose a new large-scale synthetic dataset, ObMan, with hand-object manipulations. We demonstrate the transferability of ObMan-trained models to real data.

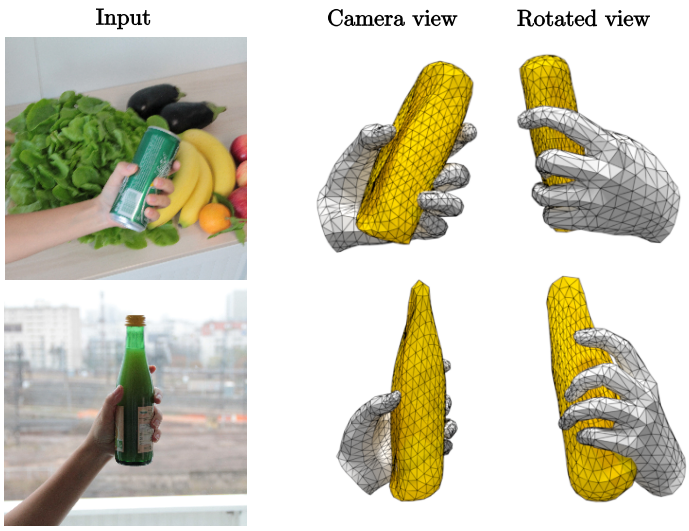


Figure B-1 – Our method jointly reconstructs hand and object meshes from a monocular RGB image. Note that the model generating the predictions for the above images, which we captured with an ordinary camera, was trained only on images from our synthetic dataset, ObMan.

B.1 Introduction

Accurate estimation of human hands, as well as their interactions with the physical world, is vital to better understand human actions and interactions. In particular, recovering the 3D shape of a hand is key to many applications including virtual and augmented reality, human-computer interaction, action recognition and imitation-based learning of robotic skills.

Hand analysis in images and videos has a long history in computer vision. Early work focused on hand estimation and tracking using articulated models [Rehg and Kanade, 1994; Heap and Hogg, 1996; Wu et al., 2001; Stenger et al., 2001] or statistical shape models [MacCormick and Isard, 2000]. The advent of RGB-D sensors brought remarkable progress to hand pose estimation from depth images [Hamer et al., 2009; Oikonomidis et al., 2011a; Keskin et al., 2012; Tang et al., 2013; Tompson et al., 2014b]. While depth sensors provide strong cues, their applicability is limited by the energy consumption and environmental constrains such as distance to the target and exposure to sunlight. Recent work obtains promising results for 2D and 3D hand pose

estimation from monocular RGB images using convolutional neural networks [Simon et al., 2017; Zimmermann and Brox, 2017; Iqbal et al., 2018; Mueller et al., 2018; Dibra et al., 2018; Spurr et al., 2018; Panteleris et al., 2018]. Most of this work, however, targets sparse keypoint estimation which is not sufficient for reasoning about hand-object contact. Full 3D hand meshes are sometimes estimated from images by fitting a hand mesh to detected joints [Panteleris et al., 2018] or by tracking given a good initialization [De La Gorce et al., 2011]. Recently, the 3D *shape* or *surface* of a hand using an end-to-end learnable model has been addressed with depth input [Malik et al., 2018].

Interactions impose constraints on relative configurations of hands and objects. For example, stable object grasps require contacts between hand and object surfaces, while solid objects prohibit penetration. In this work we exploit constraints imposed by object manipulations to reconstruct hands and objects as well as to model their interactions. We build on a parametric hand model, MANO [Romero et al., 2017], derived from 3D scans of human hands, that provides anthropomorphically valid hand meshes. We then propose a differentiable MANO network layer enabling end-to-end learning of hand shape estimation. Equipped with the differentiable shape-based hand model, we next design a network architecture for joint estimation of hand shapes, object shapes and their relative scale and translation. We also propose a novel contact loss that penalizes penetrations and encourages contact between hands and manipulated objects. An overview of our method is illustrated in Figure B-2.

Real images with ground truth shape for interacting hands and objects are difficult to obtain in practice. Existing datasets with hand-object interactions are either too small for training deep neural networks [Tzionas et al., 2016] or provide only partial 3D hand or object annotations [Sridhar et al., 2016]. The recent dataset by Garcia-Hernando et al. [2018] provides 3D hand joints and meshes of 4 objects during hand-object interactions.

Synthetic datasets are an attractive alternative given their scale and readily-

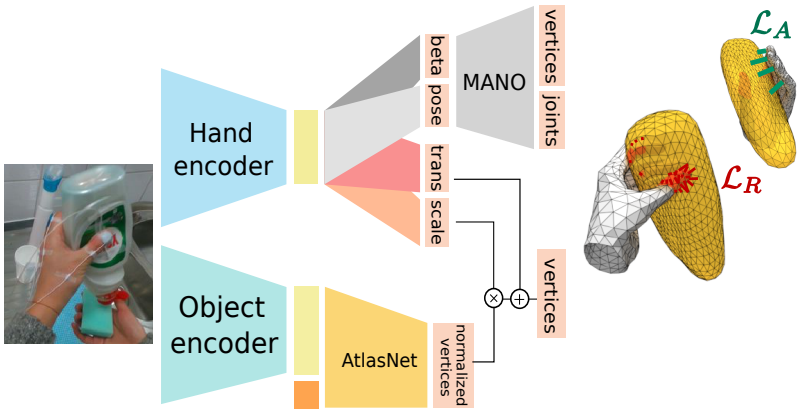


Figure B-2 – Our model predicts the hand and object meshes in a single forward pass in an end-to-end framework. The repulsion loss $\mathcal{L}_R$ penalizes interpenetration while the attraction loss $\mathcal{L}_A$ encourages the contact regions to be in contact with the object.

available ground truth. Datasets with synthesized hands have been recently introduced [Zimmermann and Brox, 2017; Mueller et al., 2018; Malik et al., 2018] but they do not contain hand-object interactions. We generate a new large-scale synthetic dataset with objects manipulated by hands: ObMan (*Ob*ject *Man*ipulation). We achieve diversity by automatically generating hand grasp poses for $2.7K$ everyday object models from 8 object categories. We adapt MANO to be able to interface it with an automatic grasp generation tool based on the GraspIt software [Miller and Allen, 2004]. ObMan is sufficiently large and diverse to support training and ablation studies of our deep models, and sufficiently realistic to generalize to real images. See Figure B-1 for reconstructions obtained for real images when training our model on ObMan.

In summary we make the following contributions. First, we design the first end-to-end learnable model for joint 3D reconstruction of hands and objects from RGB data. Second, we propose a novel contact loss penalizing penetrations and encouraging contact between hands and objects. Third, we create a new large-scale synthetic dataset, ObMan, with hand-object manipulations. The ObMan dataset and our pre-trained models and code are publicly available at [ObMan project page].

B.2 Related work

In the following, we review methods that address hand and object reconstructions in isolation. We then present related works that jointly reconstruct hand-object interactions.

Hand pose estimation. Hand pose estimation has attracted a lot of research interest since the 90*s* [Heap and Hogg, 1996; Rehg and Kanade, 1994]. The availability of commodity RGB-D sensors [Kinect; PrimeSense; Shotton et al., 2011], led to significant progress in estimating 3D hand pose given depth or RGB-D input [Hamer et al., 2009; Keskin et al., 2012]. Recently, the community has shifted its focus to RGB-based methods [Iqbal et al., 2018; Mueller et al., 2018; Panteleris et al., 2018; Simon et al., 2017; Zimmermann and Brox, 2017]. To overcome the lack of 3D annotated data, many methods employed synthetic training images [Dibra et al., 2018; Mueller et al., 2018, 2017; Zimmermann and Brox, 2017; Malik et al., 2018]. Similar to these approaches, we make use of synthetic renderings, but we additionally integrate object interactions.

3D hand pose estimation has often been treated as predicting 3D positions of *sparse* joints [Iqbal et al., 2018; Mueller et al., 2018; Zimmermann and Brox, 2017]. Unlike methods that predict only skeletons, our focus is to output a *dense* hand mesh to be able to infer interactions with objects. Very recently, Panteleris et al. [2018] and Malik et al. [2018] produce full hand meshes. However, [Panteleris et al., 2018] achieves this as a post-processing step by fitting to 2D predictions. Our hand estimation component is most similar to [Malik et al., 2018]. In contrast to [Malik et al., 2018], our method takes not depth but RGB images as input, which is more challenging and more general.

Regarding hand pose estimation in the presence of objects, Mueller et al. [2017, 2018] grasp 7 objects in a merged reality environment to render synthetic hand pose datasets. However, objects only serve the role of occluders, and the approach is difficult to scale to more object instances.

Object reconstruction. How to represent 3D objects in a CNN framework is an active research area. Voxels [Maturana and Scherer, 2015; Wu et al., 2017], point clouds [Su et al., 2017a], and more recently mesh surfaces [Groueix et al., 2018a; Kato et al., 2018; Wang et al., 2018b] have been explored. We employ the latter since meshes allow better modeling of the interaction with the hand. AtlasNet [Groueix et al., 2018a] inputs vertex coordinates concatenated with image features and outputs a deformed mesh. More recently, Pixel2Mesh [Wang et al., 2018b] explores regularizations to improve the perceptual quality of predicted meshes. Previous works mostly focus on producing accurate shape and they output the object in a normalized coordinate frame in a category-specific canonical pose. We employ a view-centered variant of [Groueix et al., 2018a] to handle generic object categories, without any category-specific knowledge. Unlike existing methods that typically input simple renderings of CAD models, such as ShapeNet [Chang et al., 2015], we work with complex images in the presence of hand occlusions.

Hand-object reconstruction. Joint reconstruction of hands and objects has been studied with multi-view RGB [Oikonomidis et al., 2011b; Ballan et al., 2012; Wang et al., 2013c] and RGB-D input with either optimization [Hamer et al., 2009, 2010; Oikonomidis et al., 2012; Tzionas et al., 2016; Tzionas and Gall, 2015; Sridhar et al., 2016; Pham et al., 2018; Tsoli and Argyros, 2018] or classification [Romero et al., 2010; Rogez et al., 2014, 2015a,b] approaches. These works use rigid objects, except for a few that use articulated [Tzionas et al., 2016] or deformable objects [Tsoli and Argyros, 2018]. Focusing on contact points, most works employ proximity metrics [Sridhar et al., 2016; Tzionas et al., 2016; Tsoli and Argyros, 2018], while [Rogez et al., 2015b] directly regresses them from images, and [Pham et al., 2018] uses contact measurements on instrumented objects. [Tzionas et al., 2016] integrates physical constraints for penetration and contact, attracting fingers onto the object one-directionally. On the contrary, [Tsoli and Argyros, 2018] symmetrically attracts the fingertips and the object surface. The last two approaches evaluate all possible configurations of contact

points and select the one that provides the most stable grasp [Tzionas et al., 2016] or best matches visual evidence [Tsoli and Argyros, 2018]. Most related to our work, given an RGB image, Romero et al. [2010] query a large synthetic dataset of rendered hands interacting with objects to retrieve configurations that match the visual evidence. Their method's accuracy, however, is limited by the variety of configurations contained in the database. In parallel work to ours [Tekin et al., 2019] jointly estimates hand skeletons and 6DOF for objects. Our work differs from previous hand-object reconstruction methods mainly by incorporating an end-to-end learnable CNN architecture that benefits from a differentiable hand model and differentiable physical constraints on penetration and contact.

B.3 Hand-object reconstruction

As illustrated in Figure B-2, we design a neural network architecture that reconstructs the hand-object configuration in a single forward pass from a rough image crop of a left hand holding an object. Our network architecture is split into two branches. The first branch reconstructs the object shape in a normalized coordinate space. The second branch predicts the hand mesh as well as the information necessary to transfer the object to the hand-relative coordinate system. Each branch has a ResNet18 [He et al., 2015] encoder pre-trained on ImageNet [Russakovsky et al., 2015]. In the following, we detail the three components of our method: hand mesh estimation in Section B.3.1, object mesh estimation in Section B.3.2, and the contact between the two meshes in Section B.3.3.

B.3.1 Differentiable hand model

Following the methods that integrate the SMPL parametric body model [Loper et al., 2015] as a network layer [Kanazawa et al., 2018a; Pavlakos et al., 2018b], we integrate the MANO hand model [Romero et al., 2017] as a differentiable layer. MANO is a

statistical model that maps pose (θ) and shape (β) parameters to a mesh. While the pose parameters capture the angles between hand joints, the shape parameters control the person-specific deformations of the hand; see [Romero et al., 2017] for more details.

Hand pose lives in a low-dimensional subspace [Romero et al., 2017; Lin et al., 2000]. Instead of predicting the full 45-dimensional pose space, we predict 30 pose PCA components. We found that performance saturates at 30 PCA components and keep this value for all our experiments (see Section B.6.2).

Supervision on vertex and joint positions ($\mathcal{L}_{V_{Hand}}, \mathcal{L}_J$). The hand encoder produces an encoding Φ_{Hand} from an image. Given Φ_{Hand}, a fully connected network regresses θ and β. We integrate the mesh generation as a differentiable network layer that takes θ and β as inputs and outputs the hand vertices V_{Hand} and 16 hand joints. In addition to MANO joints, we select 5 vertices on the mesh as fingertips to obtain 20 hand keypoints J. We define the supervision on the vertex positions ($\mathcal{L}_{V_{Hand}}$) and joint positions ($\mathcal{L}_J$) to enable training on datasets where a ground truth hand surface is not available. Both losses are defined as the L2 distance to the ground truth. We use root-relative 3D positions as supervision for $\mathcal{L}_{V_{Hand}}$ and $\mathcal{L}_J$. Unless otherwise specified, we use the wrist defined by MANO as the root joint.

Regularization on hand shape ($\mathcal{L}_\beta$). Sparse supervision can cause extreme mesh deformations when the hand shape is unconstrained. We therefore use a regularizer, $\mathcal{L}_\beta = \|\beta\|^2$, on the hand shape to constrain it to be close to the average shape in the MANO training set, which corresponds to $\beta = \vec{0} \in \mathbb{R}^{10}$.

The resulting hand reconstruction loss $\mathcal{L}_{Hand}$ is the summation of all $\mathcal{L}_{V_{Hand}}$, $\mathcal{L}_J$ and $\mathcal{L}_\beta$ terms:

$$\mathcal{L}_{Hand} = \mathcal{L}_{V_{Hand}} + \mathcal{L}_J + \mathcal{L}_\beta. \tag{B.1}$$

Our experiments indicate benefits for all three terms. Our hand branch also matches state-of-the-art performance on a standard benchmark for 3D hand pose estimation (see Section B.6.3).

B.3.2 Object mesh estimation

Following recent methods [Kato et al., 2018; Wang et al., 2018b], we focus on genus 0 topologies. We use AtlasNet [Groueix et al., 2018a] as the object prediction component of our neural network architecture. AtlasNet takes as input the concatenation of point coordinates sampled either on a set of square patches or on a sphere, and image features Φ_{Obj}. It uses a fully connected network to output new coordinates on the surface of the reconstructed object. AtlasNet explores two sampling strategies: sampling points from a sphere and sampling points from a set of squares. Preliminary experiments showed better generalization to unseen classes when input points were sampled on a sphere. In all our experiments we deform an icosphere of subdivision level 3 which has 642 vertices. AtlasNet was initially designed to reconstruct meshes in canonical view. In our model, meshes are reconstructed in view-centered coordinates. We experimentally verified that AtlasNet can accurately reconstruct meshes in this setting (see Section B.7.1). Following AtlasNet, the supervision for object vertices is defined by the symmetric Chamfer loss between the predicted vertices and points randomly sampled on the ground-truth external surface of the object.

Regularization on object shape ($\mathcal{L}_E, \mathcal{L}_L$). In order to reason about the inside and outside of the object, it is important to predict meshes with well-defined surfaces and good quality triangulations. However AtlasNet does not explicitly enforce constraints on mesh quality. We find that when learning to model a limited number of object shapes, the triangulation quality is preserved. However, when training on the larger variety of objects of ObMan, we find additional regularization on the object meshes beneficial. Following [Wang et al., 2018b; Kanazawa et al., 2018b; Groueix et al., 2018b] we employ two losses that penalize irregular meshes. We penalize edges with lengths different from the average edge length with an edge-regularization loss, $\mathcal{L}_E$. We further introduce a curvature-regularizing loss, $\mathcal{L}_L$, based on [Kanazawa et al., 2018b], which encourages the curvature of the predicted mesh to be similar to the curvature of a sphere (see details in Section B.7.2). We balance the weights of $\mathcal{L}_E$ and

$\mathcal{L}_L$ by weights μ_E and μ_L respectively, which we empirically set to 2 and 0.1. These two losses together improve the quality of the predicted meshes, as we show in Figure A.4 of the appendix. Additionally, when training on the ObMan dataset, we first train the network to predict normalized objects, and then freeze the object encoder and the AtlasNet decoder while training the hand-relative part of the network. When training the objects in normalized coordinates, noted with n, the total object loss is:

$$\mathcal{L}^n_{Object} = \mathcal{L}^n_{V_{Obj}} + \mu_L \mathcal{L}_L + \mu_E \mathcal{L}_E. \tag{B.2}$$

Hand-relative coordinate system ($\mathcal{L}_S, \mathcal{L}_T$). Following AtlasNet [Groueix et al., 2018a], we first predict the object in a normalized scale by offsetting and scaling the ground truth vertices so that the object is inscribed in a sphere of fixed radius. However, as we focus on hand-object interactions, we need to estimate the object position and scale relative to the hand. We therefore predict translation and scale in two branches, which output the three offset coordinates for the translation (i.e., x, y, z) and a scalar for the object scale. We define $\mathcal{L}_T = \|T - \hat{T}\|_2^2$ and $\mathcal{L}_S = \|S - \hat{S}\|_2^2$, where $\hat{T}$ and $\hat{S}$ are the predicted translation and scale. T is the ground truth object centroid in hand-relative coordinates and S is the ground truth maximum radius of the centroid-centered object.

Supervision on object vertex positions ($\mathcal{L}^n_{V_{Obj}}, \mathcal{L}_{V_{Obj}}$). We multiply the AtlasNet decoded vertices by the predicted scale and offset them according to the predicted translation to obtain the final object reconstruction. Following AtlasNet, the supervision for object vertices is defined by the symmetric Chamfer loss between the predicted vertices and points randomly sampled on the ground-truth external surface of the object. Chamfer loss ($\mathcal{L}_{V_{Obj}}$) is applied after translation and scale are applied. When training in hand-relative coordinates the loss becomes:

$$\mathcal{L}_{Object} = \mathcal{L}_T + \mathcal{L}_S + \mathcal{L}_{V_{Obj}}. \tag{B.3}$$

B.3.3 Contact loss

So far, the prediction of hands and objects does not leverage the constraints that guide objects interacting in the physical world. Specifically, it does not account for our prior knowledge that objects can not interpenetrate each other and that, when grasping objects, contacts occur at the surface between the object and the hand. We formulate these contact constraints as a differentiable loss, $\mathcal{L}_{Contact}$, which can be directly used in the end-to-end learning framework. We incorporate this additional loss using a weight parameter μ_C, which we set empirically to 10.

We rely on the following definition of distances between points. $d(v, V_{Obj}) = \inf_{w \in V_{Obj}} \|v - w\|_2$ denotes distances from point to set and $d(C, V_{Obj}) = \inf_{v \in C} d(v, V_{Obj})$ denotes distances from set to set. Moreover, we define a common penalization function $l_\alpha(x) = \alpha \tanh\left(\frac{x}{\alpha}\right)$, where α is a characteristic distance of action.

Repulsion ($\mathcal{L}_R$). We define a repulsion loss ($\mathcal{L}_R$) that penalizes hand and object *interpenetration*. To detect interpenetration, we first detect hand vertices that are inside the object. Since the object is a deformed sphere, it is watertight. We therefore cast a ray from the hand vertex and count the number of times it intersects the object mesh to determine whether it is inside or outside the predicted mesh [Möller and Trumbore, 1997]. $\mathcal{L}_R$ affects all hand vertices that belong to the interior of the object, which we denote $\text{Int}(Obj)$. The repulsion loss is defined as:

$$\mathcal{L}_R(V_{Obj}, V_{Hand}) = \sum_{v \in V_{Hand}} \mathbb{1}_{v \in \text{Int}(V_{Obj})} l_r(d(v, V_{Obj})),$$

where r is the repulsion characteristic distance, which we empirically set to 2cm in all experiments.

Attraction ($\mathcal{L}_A$). We further define an attraction loss ($\mathcal{L}_A$) to penalize cases in which hand vertices are in the vicinity of the object but the surfaces are *not* in contact. This loss is applied only to vertices which belong to the exterior of the object $\text{Ext}(Obj)$.

We compute statistics on the automatically-generated grasps described in the next

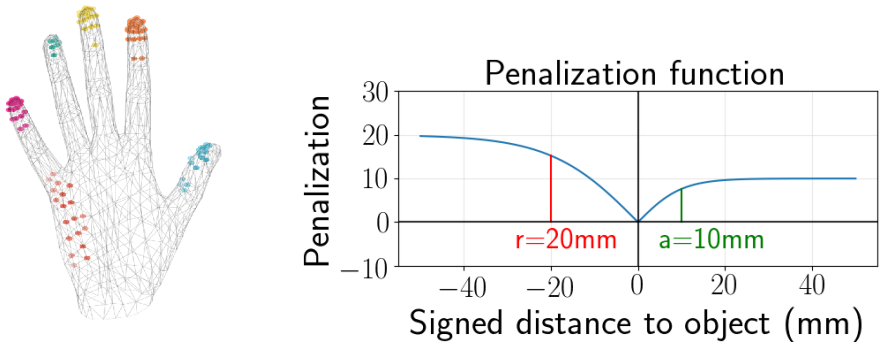


Figure B-3 – Left: Estimated contact regions from ObMan. We find that points that are often involved in contacts can be clustered into 6 regions on the palmar surface of the hand. Right: Generic shape of the penalization function emphasizing the role of the characteristic distances.

section to determine which vertices on the hand are frequently involved in contacts. We compute for each MANO vertex how often across the dataset it is in the immediate vicinity of the object (defined as less than 3mm away from the object's surface). We find that by identifying the vertices that are close to the objects in at least 8% of the grasps, we obtain 6 regions of connected vertices $\{C_i\}_{i\in[\![1,6]\!]}$ on the hand which match the 5 fingertips and part of the palm of the hand, as illustrated in Figure B-3 (left). The attraction term $\mathcal{L}_A$ penalizes distances from each of the regions to the object, allowing for sparse guidance towards the object's surface:

$$\mathcal{L}_A(V_{Obj}, V_{Hand}) = \sum_{i=1}^{6} l_a(d(C_i \cap \text{Ext}(Obj), V_{Obj})). \tag{B.4}$$

We set a to 1cm in all experiments. For regions that are further from the hand than a threshold a, the attraction will significantly decrease and become negligible as the distance to the object further increases, see Figure B-3 (right).

Our final contact loss $\mathcal{L}_{Contact}$ is a weighted sum of the attraction $\mathcal{L}_A$ and the repulsion $\mathcal{L}_R$ terms:

$$\mathcal{L}_{Contact} = \lambda_R \mathcal{L}_R + (1 - \lambda_R)\mathcal{L}_A, \tag{B.5}$$

where $\lambda_R \in [0, 1]$ is the contact weighting coefficient, e.g., $\lambda_R = 1$ means only the

repulsion term is active. We show in our experiments that the balancing between attraction and repulsion is very important for physical quality.

Our network is first trained with $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$. We then continue training with $\mathcal{L}_{Hand} + \mathcal{L}_{Object} + \mu_C \mathcal{L}_{Contact}$ to improve the physical quality of the hand-object interaction. Section B.8.1 gives further implementation details.

B.4 ObMan dataset

To overcome the lack of adequate training data for our models, we generate a large-scale synthetic image dataset of hands grasping objects which we call the *ObMan* dataset. Here, we describe how we scale automatic generation of hand-object images.

Objects. In order to find a variety of high-quality meshes of frequently manipulated everyday objects, we selected models from the ShapeNet [Chang et al., 2015] dataset. We selected 8 object categories of everyday objects (bottles, bowls, cans, jars, knifes, cellphones, cameras and remote controls). This results in a total of 2754 meshes which are split among the training, validation and test sets.

Grasps. In order to generate plausible grasps, we use the GraspIt software [Miller and Allen, 2004] following the methods used to collect the Grasp Database [Goldfeder et al., 2009]. In the robotics community, this dataset has remained valuable over many years [Sahbani et al., 2012] and is still a reference for the fast synthesis of grasps given known object models [Lenz et al., 2015; Mahler et al., 2017].

We favor simplicity and robustness of the grasp generation over the accuracy of the underlying model. The software expects a rigid articulated model of the hand. We transform MANO by separating it into 16 rigid parts, 3 parts for the phalanges of each finger, and one for the hand palm. Given an object mesh, GraspIt produces different grasps from various initializations. Following [Goldfeder et al., 2009], our generated grasps optimize for the grasp metric but do not necessarily reflect the statistical distribution of human grasps. We sort the obtained grasps according to

Figure B-4 – **ObMan:** large-scale synthetic dataset of hand-object interactions. We pose the MANO hand model [Romero et al., 2017] to grasp [Miller and Allen, 2004] a given object mesh. The scenes are rendered with variation in texture, lighting, and background.

a heuristic measure (see Section B.8.2) and keep the two best candidates for each object. We generate a total of $21K$ grasps.

Body pose. For realism, we render the hand and the full body (see Figure B-4). The pose of the hand is transferred to hands of the SMPL+H [Romero et al., 2017] model which integrates MANO to the SMPL [Loper et al., 2015; Romero et al., 2017] statistical body model, allowing us to render realistic images of embodied hands. Although we zoom our cameras to focus on the hands, we vary the body poses to provide natural occlusions and coherent backgrounds. Body poses and shapes are varied by sampling from the same distribution as in SURREAL [Varol et al., 2017]; i.e., sampling poses from the CMU MoCap database [Carnegie-Mellon Mocap Database] and shapes from CAESAR [Robinette et al., 2002]. In order to maximize the viewpoint variability, a global rotation uniformly sampled in $SO(3)$ is also applied to the body. We translate the hand root joint to the camera's optical axis. The distance to the camera is sampled uniformly between 50cm and 80cm.

Textures. Object textures are randomly sampled from the texture maps provided with ShapeNet [Chang et al., 2015] models. The body textures are obtained from the full body scans used in SURREAL [Varol et al., 2017]. Most of the scans have missing color values in the hand region. We therefore combine the body textures with

176 high resolution textures obtained from hand scans from 20 subjects. The hand textures are split so that textures from 14 subjects are used for training and 3 for test and validation sets. For each body texture, the skin tone of the hand is matched to the subject's face color. Based on the face skin color, we query in the HSV color space the 3 closest hand texture matches. We further shift the HSV channels of the hand to better match the person's skin tone.

Rendering. Background images are sampled from both the LSUN [Yu et al., 2015] and ImageNet [Russakovsky et al., 2015] datasets. We render the images using Blender [Blender - a 3D modelling and rendering package]. In order to ensure the hand and objects are visible, we discard configurations if less than 100 pixels of the hand or if less than 40% of the object is visible.

For each hand-object configuration, we render object-only, hand-only, and hand-object images, as well as the corresponding segmentation and depth maps. The dataset will be made publicly available.

In the following, we present our experimental analysis on hand-object reconstruction (Section B.5), hand-only reconstruction (Section B.6), and object-only reconstruction (Section B.7). We give implementation details in Section B.8.

B.5 Experiments on hand-object reconstruction

In this section, we first define the evaluation metrics (Section B.5.1) and the datasets (Section B.5.2) for our experiments. We then analyze the effects of occlusions (Section B.5.3) and the contact loss (Section B.5.4). Finally, we present our transfer learning experiments from synthetic to real domain (Sections B.5.5 and B.5.6).

B.5.1 Evaluation metrics

Our output is structured, and a single metric does not fully capture performance. We therefore rely on multiple evaluation metrics.

Hand error. For hand reconstruction, we compute mean end-point error (mm) over 21 joints following [Zimmermann and Brox, 2017].

Object error. Following AtlasNet [Groueix et al., 2018a], we measure the accuracy of object reconstruction by computing the symmetric Chamfer distance (mm) between points sampled on the ground-truth mesh and vertices of the predicted object mesh.

Contact. To measure the physical quality of our joint reconstruction, we use the following metrics.

Penetration depth (mm), Intersection volume (cm^3): Hands and objects should not share the same physical space. To measure whether this rule is violated, we report the intersection volume between the object and the hand as well as the penetration depth. To measure the intersection volume of the hand and object we voxelize the hand and object using a voxel size of 0.5cm. If the hand and the object collide, the penetration depth is the maximum of the distances from hand mesh vertices to the object's surface. In the absence of collision, the penetration depth is 0.

Simulation displacement (mm): Following [Tzionas et al., 2016], we use physics simulation to evaluate the quality of the produced grasps. This metric measures the average displacement of the object's center of mass in a simulated environment [Coumans, 2013] assuming the hand is fixed and the object is subjected to gravity. Details on the setup and the parameters used for the simulation can be found in [Tzionas et al., 2016]. Good grasps should be stable in simulation. However, stable simulated grasps can also occur if the forces resulting from the collisions balance each other. For estimating grasp quality, simulated displacement must be analyzed in conjunction with a measure of collision. If both displacement in simulation and penetration depth are decreasing, there is strong evidence that the physical quality of the grasp is improving (see Section B.5.4 for an analysis). The reported metrics are averaged across the dataset.

B.5.2 Datasets

In the following, we present the datasets we use to evaluate our models. Statistics for each dataset are summarized in Table B.1.

First-person hand benchmark (FHB). This dataset [Garcia-Hernando et al., 2018] is a recent video collection providing 3D hand annotations for a wide range of hand-object interactions. The joints are automatically annotated using magnetic sensors strapped on the hands, and which are visible on the RGB images. 3D mesh annotations are provided for four objects: three different bottles and a salt box. In order to ensure that the object being interacted with is unambiguously defined, we filter frames in which the manipulating hand is further than 1cm away from the manipulated object. We refer to this filtered dataset as FHB. As the milk bottle is a genus-1 object and is often grasped by its handle, we exclude this object from the experiments we conduct on contacts. We call this subset FHB_C. We use the same subject split as [Garcia-Hernando et al., 2018], therefore, each object is present in both the training and test splits.

The object annotations for this dataset suffer from some imprecisions. To investigate the range of the object ground-truth error, we measure the penetration depth of the hand skeleton in the object for each hand-object configuration. We find that on the training split of FHB, the average penetration depth is 11.0mm (std=8.9mm). While we still report quantitative results on objects for completeness, the ground-truth errors prevent us from drawing strong conclusions from reconstruction metric fluctuations on this dataset.

Hands in action dataset (HIC). We use a subset of the HIC dataset [Tzionas et al., 2016] which has sequences of a single hand interacting with objects. This gives us 4 sequences featuring manipulation of a sphere and a cube. We select the frames in which the hand is less than 5mm away from the object. We split this dataset into 2 training and 2 test sequences with each object appearing in both splits and restrict our predictions to the frames in which the minimal distance between hand and object

Table B.1 – Dataset details for train/test splits.

	Datasets			
	ObMan	FHB	FHB_C	HIC
#frames	$70K/6K$	8420/9103	5077/5657	251/307
#video sequences	-	115/127	76/88	2/2
#object instances	1947/411	4	3	2
real	no	yes	yes	yes

Table B.2 – We first show that training with occlusions is important when targeting images of hand-object interactions.

	Evaluation images			Evaluation images	
Training	H-img	HO-img	Training	O-img	HO-img
H-img ($\mathcal{L}_H$)	**10.3**	14.1	O-img ($\mathcal{L}_O$)	**0.0242**	0.0722
HO-img ($\mathcal{L}_H$)	11.7	**11.6**	HO-img ($\mathcal{L}_O$)	0.0319	**0.0302**

vertices is below 5mm. For this dataset the hand and object meshes are provided. We fit MANO to the provided hand mesh, allowing for dense point supervision on both hands and objects.

B.5.3 Effect of occlusions

For each sample in our synthetic dataset, in addition to the hand-object image (HO-img) we render two images of the corresponding isolated and unoccluded hand (H-img) or object (O-img). With this setup, we can systematically study the effect of

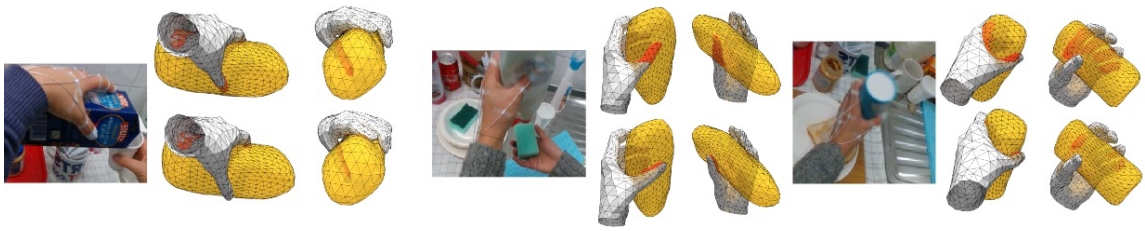


Figure B-5 – Qualitative comparison between *with* (bottom) and *without* (top) contact on FHB_C. Note the improved contact and reduced penetration, highlighted with red regions, with our contact loss.

	ObMan Dataset					FHB_C Dataset				
	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection Volume
No contact loss	11.6	641.5	9.5	31.3	12.3	28.1 ± 0.5	1579.2 ± 66.2	18.7 ±0.6	51.2 ± 1.7	26.9 ± 0.2
Only attraction ($\lambda_R = 0$)	11.9	637.8	11.8	26.8	17.4	28.4 ± 0.6	1586.9 ± 58.3	22.7 ±0.7	48.5 ± 3.2	41.2 ± 0.3
Only repulsion ($\lambda_R = 1$)	12.0	639.0	6.4	38.1	8.1	28.6 ± 0.8	1603.7 ± 49.9	6.0 ± 0.3	53.9 ± 2.3	7.1 ± 0.1
Attraction + Repulsion ($\lambda_R = 0.5$)	11.6	637.9	9.2	30.9	12.2	28.8 ±0.8	1565.0 ± 65.9	12.1 ± 0.7	47.7 ±2.5	17.6 ± 0.2

Table B.3 – We experiment with each term of the contact loss. Attraction ($\mathcal{L}_A$) encourages contacts between close points while repulsion ($\mathcal{L}_R$) penalizes interpenetration. λ_R is the repulsion weight, balancing the contribution of the two terms.

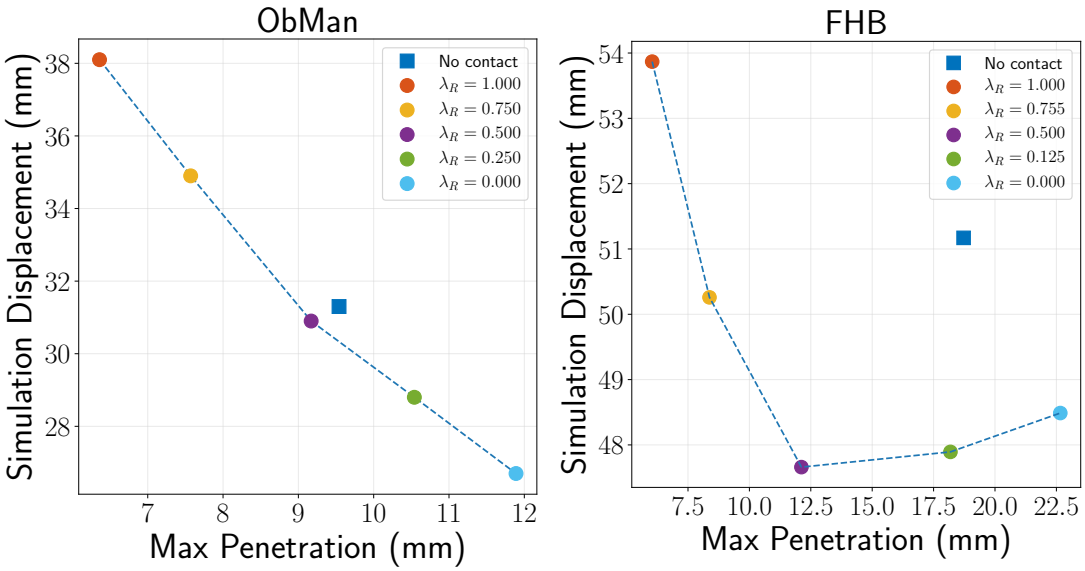


Figure B-6 – We examine the relative importance between the contact terms on the grasp quality metrics. Introducing a well-balanced contact loss improves upon the baseline on both max penetration and simulation displacement.

occlusions on ObMan, which would be impractical outside of a synthetic setup.

We study the effect of objects occluding hands by training two networks, one trained on hand-only images and one on hand-object images. We report performance on both unoccluded and occluded images. A symmetric setup is applied to study the effect of hand occlusions on objects. Since the hand-relative coordinates are not applicable for experiments with object-only images, we study the normalized shape reconstruction, centered on the object centroid, and scaled to be inscribed in a sphere of radius 1.

Unsurprisingly, the best performance is obtained when both training and testing on unoccluded images as shown in Table B.2. When both training and testing on occluded images, reconstruction errors for hands and objects drop significantly, by 12% and 25%, respectively. This validates the intuition that estimating hand pose and object shape in the presence of occlusions is a harder task.

We observe that for both hands and objects, the most challenging setting is training on unoccluded images while testing on images with occlusions. This shows that training with occlusions is crucial for accurate reconstruction of hands-object configurations.

B.5.4 Effect of contact loss

In the absence of explicit physical constraints, the predicted hands and objects have an average penetration depth of 9mm for ObMan and 19mm for FHB_C (see Table B.3). The presence of interpenetration at test time shows that the model is not implicitly learning the physical rules governing hand-object manipulation. The differences in physical metrics between the two datasets can be attributed to the higher reconstruction accuracy for ObMan but also to the noisy object ground truth in FHB_C which produces penetrated and likely unstable 'ground-truth' grasps.

In Figure B-6, we study the effect of introducing our contact loss as a fine-tuning step. We linearly interpolate λ_R in $[\![0,\ 1]\!]$ to explore various relative weightings of

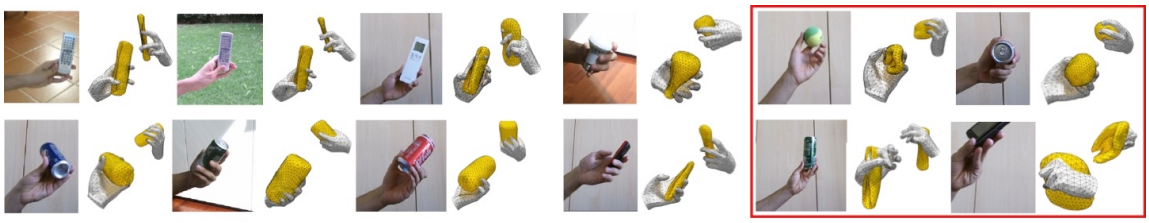


Figure B-7 – Qualitative results on CORe50. Our model, trained only on synthetic data, shows robustness to various hand poses, objects and scenes. Global hand pose and object outline are well estimated while fine details are missed. We present failure cases in the red box.

the attraction and repulsion terms. We find that using $\mathcal{L}_R$ in isolation efficiently minimizes the maximum penetration depth, reducing it by 33% for ObMan and 68% for FHB_C. This decrease occurs at the expense of the stability of the grasp in simulation. Symmetrically, $\mathcal{L}_A$ stabilizes the grasps in simulation, but produces more collisions between hands and objects. We find that equal weighting of both terms ($\lambda_R = 0.5$) improves *both* physical measures without negatively affecting the reconstruction metrics on both the synthetic and the real datasets, as is shown in Table B.3 (last row). For FHB_C, for each metric we report the means and standard deviations for 10 random seeds.

We find that on the synthetic dataset, decreased penetration is systematically traded for simulation instability whereas for FHB_C increasing λ_R from 0 to 0.5 decreases depth penetration *without* affecting the simulation stability. Furthermore, for $\lambda_R = 0.5$, we observe significant qualitative improvements on FHB_c as seen in Figure B-5.

B.5.5 Synthetic to real transfer

Large-scale synthetic data can be used to pre-train models in the absence of suitable real datasets. We investigate the advantages of pre-training on ObMan when targeting FHB and HIC. We investigate the effect of scarcity of real data on FHB by comparing pairs of networks trained using subsets of the real dataset. One is pre-trained on

ObMan while the other is initialized randomly, with the exception of the encoders, which are pre-trained on ImageNet [Russakovsky et al., 2015]. For these experiments we do not add the contact loss and report means and standard deviations for 5 distinct random seeds. We find that pre-training on ObMan is beneficial in low data regimes, especially when less than 1000 images from the real dataset are used for fine-tuning, see Figure B-8.

The HIC training set consists of only 250 images. We experiment with pre-training on variants of our synthetic dataset. In addition to ObMan, to which we refer as (a) in Figure B-9, we render $20K$ images for two additional synthetic datasets, (b) and (c), which leverage information from the training split of HIC (d). We create (b) using our grasping tool to generate automatic grasps for each of the object models of HIC and (c) using the object and pose distributions from the training split of HIC. This allows to study the importance of sampling hand-object poses from the target distribution of the real data. We explore training on (a), (b), (c) with and without fine-tuning on HIC. We find that pre-training on all three datasets is beneficial for hand and object reconstructions. The best performance is obtained when pre-training on (c). In that setup, object performance outperforms training only on real images even *before* fine-tuning, and significantly improves upon the baseline after. Hand pose error saturates after the pre-training step, leaving no room for improvement using the real data. These results show that when training on synthetic data, similarity to the target real hand-object pose distribution is critical.

B.5.6 Qualitative results on CORe50

FHB is a dataset with limited backgrounds, visible magnetic sensors and a very limited number of subjects and objects. In this section, we verify the ability of our model trained on ObMan to generalize to real data *without* fine-tuning. CORe50 [Lomonaco and Maltoni, 2017] is a dataset which contains hand-object interactions with an emphasis on the variability of objects and backgrounds. However no 3D hand or object

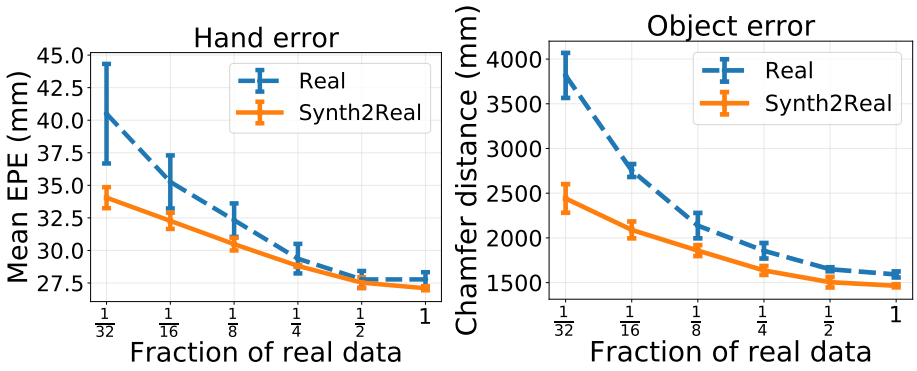


Figure B-8 – We compare training only on FHB (Real) and pre-training on synthetic, followed by fine-tuning on FHB (Synth2Real). As the amount of real data decreases, the benefit of pre-training increases. For both the object and the hand reconstruction, synthetic pre-training is critical in low-data regimes.

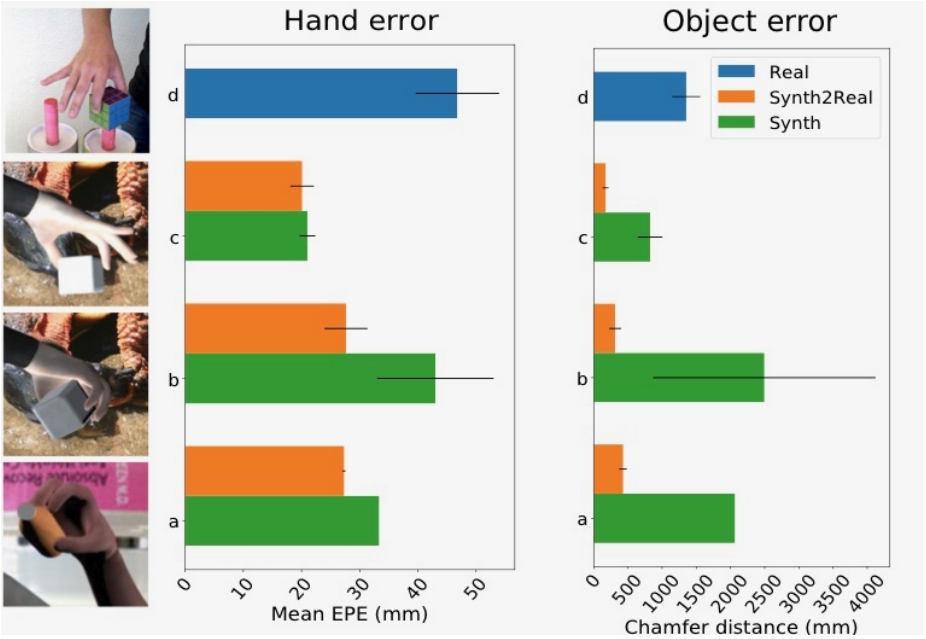


Figure B-9 – We compare the effect of training with and without fine-tuning on variants of our synthetic dataset on HIC. We illustrate each dataset (a, b, c, d) with an image sample, see text for definitions. Synthetic pre-training, whether or not the target distribution is matched, is always beneficial.

annotation is available. We therefore present qualitative results on this dataset. Figure B-7 shows that our model generalizes across different object categories, including *light-bulb*, which does not belong to the categories our model was trained on. The global outline is well recovered in the camera view while larger mistakes occur in the perpendicular direction. More results can be found in Section B.9.

B.6 Experiments on hand pose estimation

In this section, we first present an ablation study for the different losses we defined on the MANO hand model (Section B.6.1). Then, we study the latent hand representation (Section B.6.2). Finally, we validate our hand pose estimation branch and demonstrate its competitive performance compared to the state-of-the-art methods on a benchmark dataset (Section B.6.3).

B.6.1 Loss study on MANO

As explained in Section B.3.1, we define three losses for the differentiable hand model while training our network: (i) vertex positions $\mathcal{L}_{V_{Hand}}$, (ii) joint positions $\mathcal{L}_J$, and (iii) shape regularization $\mathcal{L}_\beta$. The shape is only predicted in the presence of $\mathcal{L}_\beta$. In the absence of shape regularization, when only sparse keypoint supervision is provided, predicting β without regularizing it produces extreme deformations of the hand mesh, and we therefore fix β to the average hand shape.

Table B.4 summarizes the contribution of each of these losses. Note that the dense vertex supervision is available on our synthetic dataset ObMan, and not available on the real datasets FHB [Garcia-Hernando et al., 2018] and StereoHands [Zhang et al., 2016].

We find that predicting β while regularizing it with $\mathcal{L}_\beta$ significantly improves the mean end-point-error on keypoints. On the synthetic dataset ObMan, we find that adding $\mathcal{L}_V$ yields a small additional improvement. We therefore use all three losses

Table B.4 – We report the hand reconstruction error to study different losses defined on MANO. We experiment with the loss on 3D vertices ($\mathcal{L}_{V_{Hand}}$), 3D joints ($\mathcal{L}_J$), and shape regularization ($\mathcal{L}_\beta$). We show the results of training and testing on our synthetic ObMan dataset, as well as the real datasets FHB [Garcia-Hernando et al., 2018] and StereoHands [Zhang et al., 2016].

	ObMan	FHB	StereoHands
$\mathcal{L}_J$	13.5	28.1	11.4
$\mathcal{L}_J + \mathcal{L}_\beta$	11.7	**26.5**	**10.0**
$\mathcal{L}_{V_{Hand}}$	14.0	-	-
$\mathcal{L}_{V_{Hand}} + \mathcal{L}_\beta$	12.0	-	-
$\mathcal{L}_{V_{Hand}} + \mathcal{L}_J + \mathcal{L}_\beta$	**11.6**	-	-

Table B.5 – We report the hand reconstruction error on multiple datasets to study the effect of the number of PCA hand pose components for the latent MANO representation.

#PCA comps.	6	15	30	45
FHB	28.2	27.5	**26.5**	26.9
StereoHands	13.9	11.1	**10.0**	**10.0**
ObMan	23.4	13.3	11.6	**11.2**

whenever dense vertex supervision is available, and $\mathcal{L}_J$ in conjunction with $\mathcal{L}_\beta$ when only keypoint supervision is provided.

B.6.2 MANO pose representation

As described in Section B.3.1, our hand branch outputs a 30-dimensional vector to represent the hand. These are the 30 first PCA components from the 45-dimensional full pose space. We experiment with different dimensionality for the latent hand representation and summarize our findings in Table B.5. While low-dimensionality fails to capture some poses present in the datasets, we do not observe improvements after increasing the dimensionality more than 30. Therefore, we use this value for all experiments.

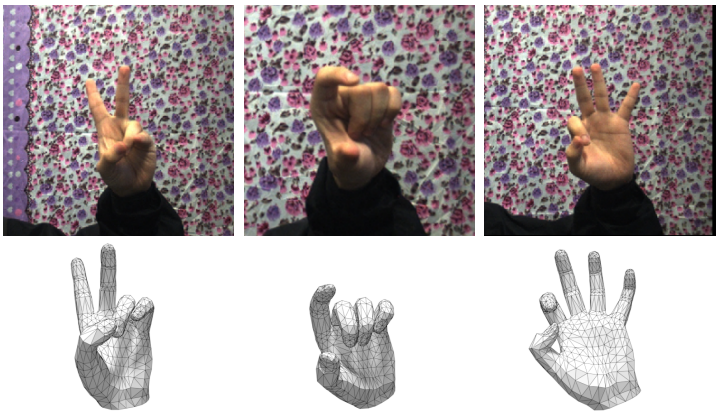


Figure B-10 – Qualitative results on the test sequence of the StereoHands dataset.

B.6.3 Comparison with the state of the art

Using the MANO branch of the network, we can also estimate the hand pose for images in which the hands are not interacting with objects, and compare our results with previous methods. We train and test on the StereoHands dataset [Zhang et al., 2016], and follow the evaluation protocol of [Iqbal et al., 2018; Zimmermann and Brox, 2017; Mueller et al., 2018] by training on 10 sequences from StereoHands and testing on the 2 remaining ones. For fair comparison, we add a palm joint to the MANO model by averaging the positions of two vertices on the front and back of the hand model at the level of the palm. Although the hand shape parameter β allows to capture the variability of hand shapes which occurs naturally in human populations, it does not account for the discrepancy between different joint conventions. To account for skeleton mismatch, we add a linear layer initialized to identity which maps from the MANO joints to the final joint annotations.

We report the area under the curve (auc) on the percentage of correct keypoints (PCK). Figure B-11 shows that our differentiable hand model is on par with the state of the art. Note that the StereoHands benchmark is close to saturation. In contrast to other methods [Cai et al., 2018; Iqbal et al., 2018; Mueller et al., 2018; Zimmermann and Brox, 2017; Sun et al., 2015b] that only predicts sparse skeleton keypoints, our model produces a *dense* hand mesh. Figure B-10 presents some qualitative results

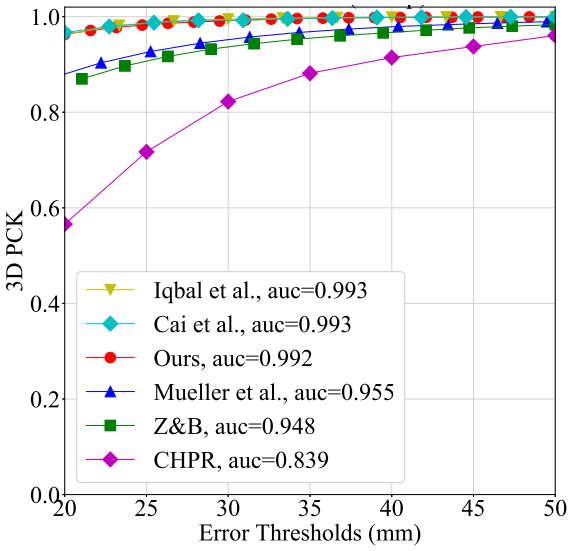


Figure B-11 – We compare our root-relative 3D hand pose estimation on StereoHands to the state-of-the-art methods from Iqbal et al. [2018], Cai et al. [2018], Mueller et al. [2018], Zimmermann and Brox [2017], and CHPR [Sun et al., 2015b].

from this dataset.

B.7 Experiments on object reconstruction

In the following, we validate our design choices for the object reconstruction branch. We experiment with object reconstruction (i) in the camera viewpoint (Section B.7.1) and (ii) with regularization losses (Section B.7.2).

B.7.1 Canonical versus camera view reconstruction

As explained in Section B.3.2, we perform object reconstructions in the camera coordinate frame. To validate that AtlasNet [Groueix et al., 2018a] can successfully predict objects in camera view as well as in canonical view, we reproduce the training setting of the original paper [Groueix et al., 2018a]. We use the setting where 2500 points are sampled on a sphere and train on the rendered images from ShapeNet [Choy et al., 2016]. To obtain the rotated reference for the object, we

Table B.6 – Chamfer loss (×2000) for 2500 points in the canonical view and camera view show no degradation from predicting the camera view reconstruction. We compare our re-implementation to the results provided by [Groueix et al., 2018a] on their code page https://github.com/ThibaultGROUEIX/AtlasNet.

	Object error
Canonical view [Groueix et al., 2018a]	4.87
Canonical view (ours)	4.88
Camera view (ours)	4.88

apply the ground truth azimuth and elevation provided with the renderings so that the 3D ground truth matches the camera view. We use the original hyperparameters (Adam [Kingma and Ba, 2014] with a learning rate of 0.001) and train both networks for 25 epochs. Both for supervision and evaluation metrics, we report the Chamfer distance $\mathcal{L}_{V_{Obj}} = \frac{1}{2}(\sum_p min_q \|p-q\|_2^2 + \sum_q min_p \|q-p\|_2^2)$ where q spans the predicted vertices and p spans points uniformly sampled on the surface of the ground truth object. We always sample the same number of points on the surface as there are vertices in the predicted mesh. We find that both numerically and qualitatively the performance is comparable for the two settings. Some reconstructed meshes in camera view are shown in Figure B-12. For better readability they also multiply the Chamfer loss by 1000. In order to provide results directly comparable with the original paper [Groueix et al., 2018a], we also report numbers with the same scaling in Table B.6. Table B.6 reports the Chamfer distances for their released model, our reimplementation in canonical view, and our implementation in non-canonical view. We find that our implementation allows us to train a model with similar performances to the released model. We observe no numerical or qualitative loss in performance when predicting the camera view instead of the canonical one.

B.7.2 Object mesh regularization

We find that in the absence of explicit regularization on their quality, the predicted meshes can be very irregular. Sharp discontinuities in curvature occur in regions

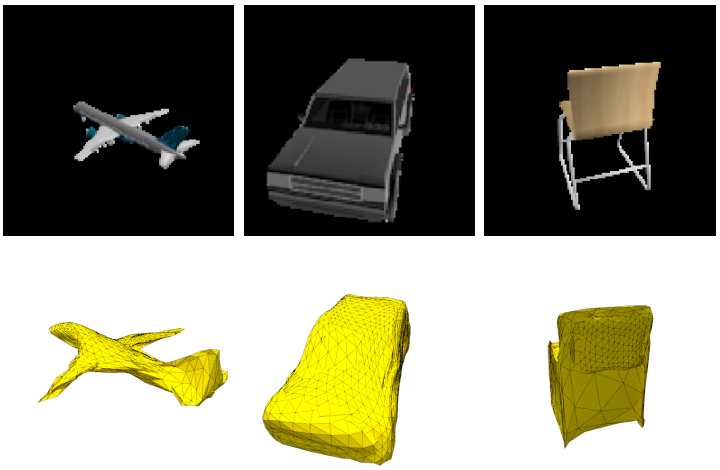


Figure B-12 – Renderings from ShapeNet models and our corresponding reconstructions in camera view.

where the ground truth mesh is smooth, and the mesh triangles can be of very different dimensions. These shortcomings can be observed on all three reconstructions in Figure B-12. Following recent work on mesh estimation from image inputs [Wang et al., 2018b; Kanazawa et al., 2018b; Groueix et al., 2018b], we introduce regularization terms on the object mesh.

Laplacian smoothness regularization ($\mathcal{L}_L$). In order to avoid unwanted discontinuities in the curvature of the mesh, we enforce a local prior of smoothness. We use the discrete Laplace-Beltrami operator to estimate the curvature at each mesh vertex position, as we have no prior on the final shape of the geometry, we compute the graph laplacian L on our mesh, which only takes into account adjacency between mesh vertices. Multiplying the laplacian L by the positions of the object vertices $\mathcal{V}_{Obj}$ produces vectors which have the same direction as the vertex normals and their norm proportional to the curvature. Minimizing the norm of these vector therefore minimizes the curvature. We minimize the mean curvature over all vertices in order to encourage smoothness on the mesh.

Laplacian edge length regularization ($\mathcal{L}_E$). $\mathcal{L}_E$ penalizes configurations in which

the edges of the mesh have different lengths. The edge regularization is defined as:

$$\mathcal{L}_E = \frac{1}{|\mathcal{E}_L|} \sum_{l \in \mathcal{E}_L} |l^2 - \mu(\mathcal{E}_L^2)|, \tag{B.6}$$

where $\mathcal{E}_L$ is the set of edge lengths, defined as the L2 norms of the edges, and $\mu(\mathcal{E}_L^2)$ is the average of the square of edge lengths.

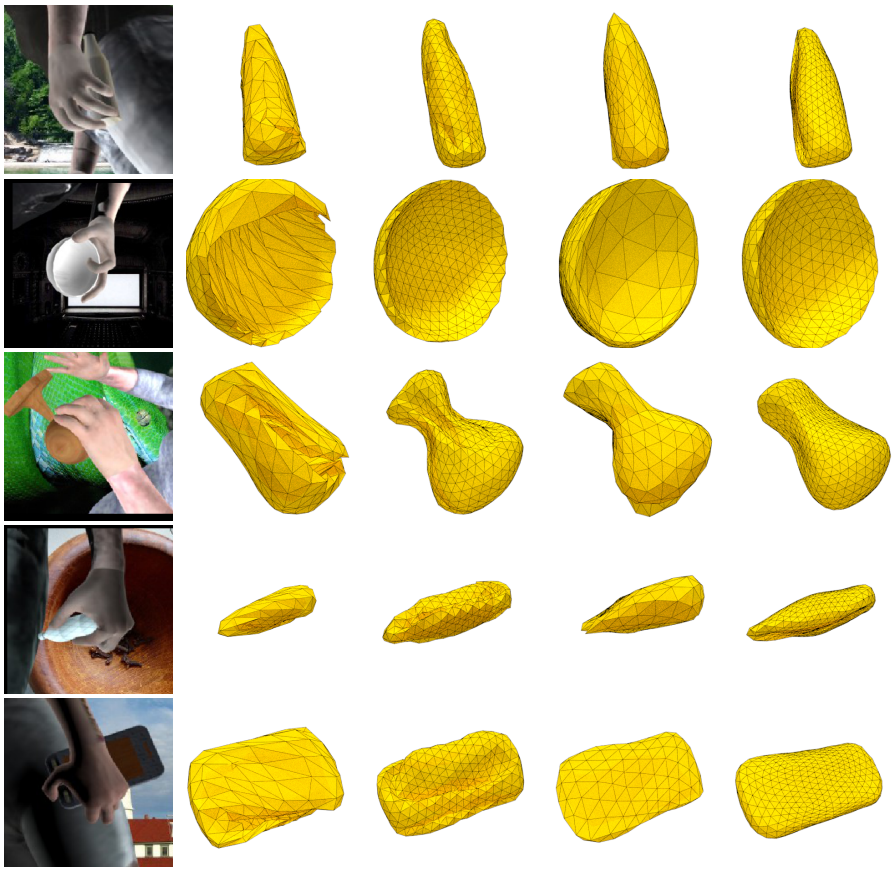
To evaluate the effect of the two regularization terms we train four different models. We train a model without any regularization, two models for which only one of the two regularization terms are active, and finally a model for which the two regularization terms are applied simultaneously. Each of these models is trained for 200 epochs.

Figure B-13 shows the qualitative benefits of each term. While edge regularization $\mathcal{L}_E$ alone already significantly improves the quality of the predicted mesh, note that unwanted bendings of the mesh still occur, for instance in the last row for the cellphone reconstruction. Adding the laplacian smoothness $\mathcal{L}_L$ resolves these irregularities.

However, adding each regularization term negatively affects the final reconstruction score. Particularly we observe that introducing edge regularization increases the Chamfer loss by 22% while significantly improving the perceptual quality of the predicted mesh. Introducing the regularization terms contributes to the coarseness of the object reconstructions, as can be observed on the third row, where sharp curvatures of the object in the input image are not captured in the reconstruction.

B.8 Implementation details

We give implementation details on our training procedure (Section B.8.1) and our automatic grasp generation (Section B.8.2).



	No reg.	$\mathcal{L}_E$	$\mathcal{L}_L$	$\mathcal{L}_E + \mathcal{L}_L$
Object error	0.0246	0.0286	0.0258	0.0292

Figure B-13 – We show the benefits from each term of the regularization. Using both the $\mathcal{L}_E$ and $\mathcal{L}_L$ in conjunction improves the visual quality of the predicted triangulation while preserving the shape of the object.

B.8.1 Training details

For all our experiments, we use the Adam optimizer [Kingma and Ba, 2014]. As we observe instabilities in validation curves when training on synthetic datasets, we freeze the batch normalization layers. This fixes their weights to the original values from the ImageNet [Russakovsky et al., 2015] pre-trained ResNet18 [He et al., 2015].

For the final model trained on ObMan, we first train the (normalized) object branch using $\mathcal{L}^n_{Object}$ for 250 epochs, we start with a learning rate of 10^{-4} and decrease it to 10^{-5} at epoch 200. We then freeze the object encoder and the AtlasNet decoder, as explained in Section B.3.2. We further train the full network with $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$ for 350 additional epochs, decreasing the learning rate from 10^{-4} to 10^{-5} after the first 200 epochs.

When fine-tuning from our main model trained on synthetic data to smaller real datasets, we unfreeze the object reconstruction branch.

For the FHB_C dataset, we train all the parts of the network simultaneously with the supervision $\mathcal{L}_{Hand} + \mathcal{L}_{Object}$ for 400 epochs, decreasing the learning rate from 10^{-4} to 10^{-5} at epoch 300.

When fine-tuning our models with the additional contact loss, $\mathcal{L}_{Hand} + \mathcal{L}_{Object} + \mu_C \mathcal{L}_{Contact}$, we use a learning rate of 10^{-5}. We additionally set the momentum of the Adam optimizer [Kingma and Ba, 2014] to zero, as we find that momentum affects negatively the training stability when we include the contact loss.

In all experiments, we keep the relative weights between different losses as provided in Section B.3 and normalize them so that the sum of all the weights equals 1.

B.8.2 Heuristic metric for sorting GraspIt grasps

We use GraspIt [Miller and Allen, 2004] to generate grasps for the ShapeNet object models. GraspIt generates a large variety of grasps by exploring different initial hand poses. However, some initializations do not produce good grasps. Similarly

to [Goldfeder et al., 2009] we filter the grasps in a post-processing step in order to retain grasps of good quality according to a heuristic metric we engineer for this purpose.

For each grasp, GraspIt provides two grasp quality metrics ε and v [Ferrari and Canny, 1992]. Each grasp produced by GraspIt [Miller and Allen, 2004] defines contact points between the hand and the object. Assuming rigid contacts with friction, we can compute the space of wrenches which can be resisted by the grasp: the grasp wrench space (GWS). This space is normalized with relation to the scale of the object, defined as the maximum radius of the object, centered at its center of mass. The grasp is suitable for any task that involves external wrenches that lie within the GWS. v is the volume of the 6-dimensional GWS, which quantifies the range of wrenches the grasp can resist. The GWS can further be characterized by the radius ε of the largest ball which is centered at the origin and inscribed in the grasp wrench space. ε is the maximal wrench norm that can be balanced by the contacts for external wrenches applied coming from arbitrary directions. ε belongs to $[0, 1]$ in the scale-normalized GWS, and higher values are associated with a higher robustness to external wrenches.

We require a single value to reflect the quality of the grasp in order to sort different grasps. We use the norm of the $[\varepsilon, v]$ vector in our heuristic measure of grasp quality. We find that in the grasps produced by GraspIt, power grasps, as defined by [Feix et al., 2016] in which larger surfaces of the hand and the object are in contact, are rarely produced. To allow for a larger proportion of power grasps, we use a multiplier γ_{palm} which we empirically set to 1 if the palm is not in contact and 3 otherwise. We further favor grasps in which a large number of phalanges are in contact with the object by weighting the final grasp score using N_p, the number of phalanges in contact with the object, which is computed by the software.

The final grasp quality score G is defined as:

$$G = \gamma_{palm}\sqrt{N_p}\|\varepsilon, v\|_2. \tag{B.7}$$

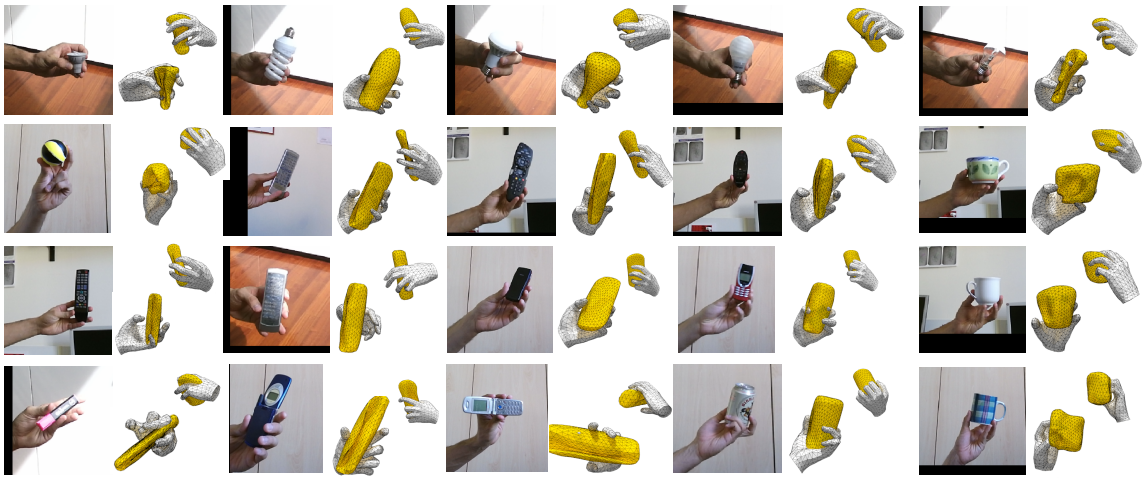


Figure B-14 – Qualitative results on CORe50 dataset. We present additional hand-object reconstructions for a variety of object categories and object instances, spanning various hand poses and object shapes.

We find that keeping the two best grasps for each object produces both diverse grasps and grasps of good quality.

B.9 Additional qualitative results

We present additional qualitative results on the CORe50 [Lomonaco and Maltoni, 2017] dataset. We present a variety of diverse input images from CORe50 in Figure B-14 alongside the predictions of our final model trained solely on ObMan.

The first row presents results on various shapes of light bulbs. Note that this category is not included in the synthetic object models of ObMan. Our model can therefore generalize across object categories. The last column shows some reconstructions of mugs, showcasing the topological limitations of the sphere baseline of AtlasNet which cannot, by construction, capture handles.

However, we observe that the object shapes are often coarse, and that fine details such as phone antennas are not reconstructed. We also observe errors in the relative position between the object and the hand, which is biased towards predicting the object's centroid in the palmar region of the hand, see Figure B-14, fourth column. As

hard constraints on collision are not imposed, hand-object interpenetration occurs in some configurations, for instance in the top-right example. In the bottom-left example we present a failure case where the hand pose violates anatomical constraints. Note that while our model predicts hand pose in a low-dimensional space, which implicitly regularizes hand poses, anatomical validity is not guaranteed.

B.10 Conclusions

We presented an end-to-end approach for joint reconstruction of hands and objects given a single RGB image as input. We proposed a novel contact loss that enforces physical constraints on the interaction between the two meshes. Our results and the ObMan dataset open up new possibilities for research on modeling object manipulations. Future directions include learning grasping affordances from large-scale visual data, and recognizing complex and dynamic hand actions.

Bibliography

Aggarwal, J. and Cai, Q. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428 – 440, 1999. 29, 30

Akhter, I. and Black, M. J. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*, 2015. 22

Alldieck, T., Kassubeck, M., Wandt, B., Rosenhahn, B., and Magnor, M. Optical flow-based 3D human motion estimation from monocular video. In *GCPR*, 2017. 26

Amit, Y. and Geman, D. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997. 19

Andriluka, M., Roth, S., and Schiele, B. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009. 19

Andriluka, M., Roth, S., and Schiele, B. Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, 99:259 – 280, 2012. 19

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 20, 25, 38, 39, 48, 58, 71, 74

Anguelov, D. *Learning Models of Shape from 3D Range Data*. PhD thesis, Stanford University, 2005. 27

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. SCAPE: Shape completion and animation of people. In *SIGGRAPH*, 2005. 26, 65

Badler, N. I., O'Rourke, J., and Tolzis, H. A human body modelling system for motion studies. In *IEEE*, volume 11, pages 1397 – 1403, 1979. 18

Badler, N. *Temporal Scene Analysis: Conceptual Descriptions of Object Movements*. PhD thesis, University of Toronto, 1975. 29, 30

Balan, A., Sigal, L., Black, M. J., Davis, J., and Haussecker, H. Detailed human shape and pose from images. In *CVPR*, 2007. 26, 64

Ballan, L., Taneja, A., Gall, J., Van Gool, L., and Pollefeys, M. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012. 176

Ballas, N., Yao, L., Pal, C. J., and Courville, A. C. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 35

Baradel, F., Wolf, C., and Mille, J. Pose-conditioned spatio-temporal attention for human action recognition. *CoRR*, abs/1703.10106, 2017. 125, 139

Baradel, F., Wolf, C., Mille, J., and Taylor, G. W. Glimpse clouds: Human activity recognition from unstructured feature points. In *CVPR*, 2018. 125, 131, 138, 139, 141, 143

Barbosa, I. B., Cristani, M., Caputo, B., Rognhaugen, A., and Theoharis, T. Looking beyond appearances: Synthetic training data for deep CNNs in re-identification. *Computer Vision and Image Understanding*, 167:50 – 62, 2018. 74

Baumberg, A. and Hogg, D. Efficient method for contour tracking using active shape models. In *Motion of Non-Rigid and Articulated Obgects Workshop*, 1994. 31

Baumberg, A. and Hogg, D. C. Generating spatiotemporal models from examples. *Image and Vision Computing*, 14:525–532, 1996. 31

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. Dynamic image networks for action recognition. In *CVPR*, 2016. 35, 100

Black, M. J., Yacoob, Y., Jepson, A. D., and Fleet, D. J. Learning parameterized models of image motion. In *CVPR*, 1997. 32

Blender - a 3D modelling and rendering package. http://www.blender.org. 42, 185

Bobick, A. J. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 352 1358:1257–65, 1997. 10

BodyNet project page. http://www.di.ens.fr/willow/research/bodynet/. 72, 87

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. 26, 62, 64, 73, 75, 76, 80

Bregler, C. Learning and recognizing human dynamics in video sequences. In *CVPR*, 1997. 32

Bregler, C. and Malik, J. Tracking people with twists and exponential maps. In *CVPR*, 1998. 31

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a "Siamese" time delay neural network. In *NIPS*, 1993. 127

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 101, 103

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 74

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 8, 151, 154, 155

Cai, Y., Ge, L., Cai, J., and Yuan, J. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*, 2018. 196, 197

Cao, K., Rong, Y., Li, C., Tang, X., and Loy, C. C. Pose-robust face recognition via deep residual equivariant mapping. In *CVPR*, 2018. 127, 134

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017. 21, 62

Carnegie-Mellon Mocap Database. http://mocap.cs.cmu.edu/. 38, 43, 184

Carreira, J. and Zisserman, A. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. 35, 122, 128

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. Human pose estimation with iterative error feedback. *CVPR*, 2016. 20

Cédras, C. and Shah, M. Motion-based recognition a survey. *Image and Vision Computing*, 13(2):129 – 155, 1995. 29

Chang, A. X., homas A. Funkhouser, eonidas J. Guibas, Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 176, 183, 184

Charades project page. http://allenai.org/plato/charades/. 153

Charles, J., Pfister, T., Everingham, M., and Zisserman, A. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, 2013. 19

Chen, C.-H. and Ramanan, D. 3D human pose estimation = 2D pose estimation + matching. In *CVPR*, 2017. 25

Chen, D. L. and Dolan, W. B. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011. 154

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015a. 24

Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016a. 25, 46

Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., and Chen, B. Synthesizing training images for boosting human 3D pose estimation. In *3DV*, 2016b. 8, 40, 41, 74

Chen, W., Fu, Z., Yang, D., and Deng, J. Single-image depth perception in the wild. In *NIPS*, 2016c. 25, 47

Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*, 2015b. 167

Chen, X. and Yuille, A. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014. 20

Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., and Yuille, A. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014. 24, 25

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, 2016. 197

Chumpy. http://chumpy.org. 73

Coumans, E. Bullet real-time physics simulation. http://bulletphysics.org, 2013. 186

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. Visual categorization with bags of keypoints. In *ECCVW*, 2004. 33

D. Bourdev, L. and Malik, J. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 19

Dalal, N., Triggs, B., and Schmid, C. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 34

De La Gorce, M., Fleet, D. J., and Paragios, N. Model-based 3D hand pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1793–1805, 2011. 173

Deng, H., Birdal, T., and Ilic, S. PPFNet: Global context aware local features for robust 3D point matching. In *CVPR*, 2018. 63

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 8, 34, 97, 151

Deutscher, J., Blake, A., and Reid, I. Articulated body motion capture by annealed particle filtering. In *CVPR*, 2000. 31

Devlin, J., Gupta, S., Girshick, R., Mitchell, M., and Zitnick, C. L. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*, 2015. 168

Dibra, E., Jain, H., Öztireli, C., Ziegler, R., and Gross, M. HS-Nets: Estimating human body shape from silhouettes with convolutional neural networks. In *3DV*, 2016. 28

Dibra, E., Melchior, S., Wolf, T., Balkis, A., Öztireli, A. C., and Gross, M. H. Monocular RGB hand pose inference from unsupervised refinable nets. In *CVPR Workshops*, 2018. 173, 175

Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 34, 98, 99, 100

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 39

Du, Y., Wong, Y., Liu, Y., Han, F., Gui, Y., Wang, Z., Kankanhalli, M., and Geng, W. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *ECCV*, 2016. 40

Dwibedi, D., Tompson, J., Lynch, C., and Sermanet, P. Learning actionable representations from visual observations. In *IROS*, 2018. 127

Efros, A., Berg, A., Mori, G., and Malik, J. Recognizing action at a distance. In *ICCV*, 2003. 32

Eichner, M., Marin-Jimenez, M., Zisserman, A., and Ferrari, V. 2D articulated human pose estimation and retrieval in (almost) unconstrained still images. *International Journal of Computer Vision*, 99(2):190–214, 2012. 19

Eigen, D. and Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 25, 46

Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014. 25, 46, 48

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html. 24

Fanello, S. R., Keskin, C., Izadi, S., Kohli, P., Kim, D., Sweeney, D., Criminisi, A., Shotton, J., Kang, S. B., and Paek, T. Learning to be a depth camera for close-range human capture and interaction. In *SIGGRAPH*, 2014. 39

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1915–1929, 2013. 24

Farhadi, A. and Tabrizi, M. Learning to recognize activities from the wrong view point. In *ECCV*, 2008. 125

Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003. 101, 103

Feichtenhofer, C., Pinz, A., and Zisserman, A. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 35, 100, 116, 122

Feix, T., Romero, J., Schmiedmayer, H.-B., Dollar, A., and Kragic, D. The grasp taxonomy of human grasp types. *Human-Machine Systems, IEEE Transactions on*, 2016. 203

Felzenszwalb, P., McAllester, D., and Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 19

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. Object detection with discriminatively trained part based models. *Pattern Analysis and Machine Intelligence*, 32(9), 2010. 19

Felzenszwalb, P. and Huttenlocher, D. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55 – 79, 2005. 18

Fernández, C., Baiget, P., Roca, F., and Gonzàlez, J. Determining the best suited semantic events for cognitive surveillance. *Expert Systems with Applications*, 38(4): 4068 – 4079, 2011. 10

Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., and Tuytelaars, T. Modeling video evolution for action recognition. In *CVPR*, 2015. 34

Ferrari, C. and Canny, J. F. Planning optimal grasps. In *ICRA*, 1992. 203

Ferrari, V., Marin-Jimenez, M., and Zisserman, A. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 19

Ferrari, V., Marín-Jiménez, M., and Zisserman, A. 2D human pose estimation in TV shows. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 128–147. Springer, 2009. 154

Fischler, M. A. and Elschlager, R. A. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, C-22(1):67–92, 1973. 18

Forsyth, D. A. and Fleck, M. M. Body plans. In *CVPR*, 1997. 18

Fouhey, D. F., Kuo, W., Efros, A. A., and Malik, J. From lifestyle VLOGs to everyday interactions. In *CVPR*, 2018. 8

Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 40

Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 173, 187, 194, 195

Gavrila, D. M. and Davis, L. S. Towards 3-D model-based tracking and recognition of human movement: a multi-view approach. In *Int. Workshop on Face and Gesture Recognition*, 1995. 31

Gavrila, D. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82 – 98, 1999. 29

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32:1231–1237, 2013. 25

Ghezelghieh, M. F., Kasturi, R., and Sarkar, S. Learning camera viewpoint using cnn to improve 3D body pose estimation. In *3DV*, 2016. 8, 40, 74

Girdhar, R., Fouhey, D., Rodriguez, M., and Gupta, A. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 63

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 97, 99

Gkioxari, G., Hariharan, B., Girshick, R., and Malik, J. Using k-poselets for detecting people and localizing their keypoints. In *CVPR*, 2014. 19

Goldfeder, C., Ciocarlie, M. T., Dang, H., and Allen, P. K. The Columbia grasp database. In *ICRA*, 2009. 183, 203

Gong, K., Liang, X., Shen, X., and Lin, L. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017. 25

Gonzàlez, J. *Human Sequence Evaluation: The Key-frame Approach*. PhD thesis, UAB, Spain, 2004. 10

Gorban, A., Idrees, H., Jiang, Y.-G., Roshan Zamir, A., Laptev, I., Shah, M., and Sukthankar, R. THUMOS challenge: Action recognition with a large number of classes, 2015. 151, 153, 155

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12): 2247–2253, 2007. 32, 33, 152

Green, R. Spherical harmonic lighting: The gritty details. In *Archives of the Game Developers Conference*, volume 56, 2003. 44

Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018a. 63, 176, 179, 180, 186, 197, 198

Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. 3D-CODED : 3D correspondences by deep deformation. In *ECCV*, 2018b. 179, 199

Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., and Malik, J. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 8

Guan, P., Weiss, A., O. Balan, A., and Black, M. Estimating human shape and pose from a single image. In *ICCV*, 2009. 26, 64

Güler, R. A., Trigeorgis, G., Antonakos, E., Snape, P., Zafeiriou, S., and Kokkinos, I. DenseReg: Fully convolutional dense shape regression in-the-wild. In *CVPR*, 2017. 66

Güler, R. A., Neverova, N., and Kokkinos, I. DensePose: Dense human pose estimation in the wild. In *CVPR*, 2018. 25, 66, 81

Gupta, A., Martinez, J., Little, J. J., and Woodham, R. J. 3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *CVPR*, 2014. 138

Gupta, A. and Davis, L. S. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007. 154

Hamer, H., Schindler, K., Koller-Meier, E., and Van Gool, L. Tracking a hand manipulating an object. In *ICCV*, 2009. 172, 175, 176

Hamer, H., Gall, J., Weise, T., and Van Gool, L. An object-dependent hand pose prior from sparse training data. In *CVPR*, 2010. 176

Hara, K., Kataoka, H., and Satoh, Y. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *CVPR*, 2018. 35, 122, 128, 130, 138, 141

Hariharan, B., Arbeláez, P. A., Girshick, R. B., and Malik, J. Simultaneous detection and segmentation. In *ECCV*, 2014. 24

Hariharan, B., Arbeláez, P. A., Girshick, R. B., and Malik, J. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 24

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., and Schmid, C. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 11, 13

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2015. 130, 177, 202

Heap, T. and Hogg, D. Towards 3D hand tracking using a deformable model. In *International Conference on Automatic Face and Gesture Recognition*, 1996. 172, 175

Hinton, G. Using relaxation to find a puppet. In *Artificial Intelligence and Simulation of Behaviour*, 1976. 18, 20

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 34

Hogg, D. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20, 1983. 18, 19

Hongeng, S. and Nevatia, R. Multi-agent event recognition. In *ICCV*, 2001. 10

Hu, J., Zheng, W., Lai, J., and Zhang, J. Jointly learning heterogeneous features for RGB-D activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2186–2200, 2017. 138, 141

Huang, Y., Bogo, F., Lassner, C., Kanazawa, A., Gehler, P. V., Romero, J., Akhter, I., and Black, M. J. Towards accurate marker-less human shape and pose estimation over time. In *3DV*, 2017. 26

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schieke, B. DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016. 21

Ioffe, S. and Forsyth, D. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45–68, 2001. ISSN 1573-1405. 18

Ionescu, C., Fuxin, L., and Sminchisescu, C. Latent structured models for human pose estimation. In *ICCV*, 2011. 23, 45, 48, 53

Ionescu, C., Carreira, J., and Sminchisescu, C. Iterated second-order label sensitive pooling for 3D human pose estimation. In *CVPR*, 2014a. 24, 45, 53

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014b. 23, 24, 38, 40, 45, 48, 53, 64, 86

Iqbal, U., Molchanov, P., Breuel, T., Gall, J., and Kautz, J. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 173, 175, 196, 197

Isard, M. and Blake, A. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 32

Iwashita, Y., Takamine, A., Kurazume, R., and Ryoo, M. S. First-person animal activity recognition from egocentric videos. In *ICPR*, 2014. 154

Jackson, A. S., Bulat, A., Argyriou, V., and Tzimiropoulos, G. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *ICCV*, 2017. 65, 66, 67

Jhuang, H., Gall, J., Zuffi, S., Schmid, C., and Black, M. J. Towards understanding action recognition. In *ICCV*, 2013. 24

Ji, S., Xu, W., Yang, M., and Yu, K. 3D convolutional neural networks for human action recognition. In *ICML*, 2010. 34, 98, 99, 100

Ji, Y., Xu, F., Yang, Y., Shen, F., Shen, H. T., and Zheng, W. A large-scale RGB-D database for arbitrary-view human action recognition. In *ACMMM*, 2018. 124, 133, 138, 141

Johnson, S. and Everingham, M. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 20, 74

Ju, S. X., Black, M. J., and Yacoob, Y. Cardboard people: a parameterized model of articulated image motion. In *International Conference on Automatic Face and Gesture Recognition*, 1996. 32

Kan, M., Shan, S., and Chen, X. Multi-view deep network for cross-view classification. In *CVPR*, 2016. 127

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. End-to-end recovery of human shape and pose. In *CVPR*, 2018a. 23, 28, 63, 65, 75, 80, 177

Kanazawa, A., Tulsiani, S., Efros, A. A., and Malik, J. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018b. 179, 199

Kantorov, V. and Laptev, I. Efficient feature extraction, encoding, and classification for action recognition. In *CVPR*, 2014. 101, 103

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 8, 34, 98, 99, 100, 116, 151, 153, 155, 164

Kato, H., Ushiku, Y., and Harada, T. Neural 3D mesh renderer. In *CVPR*, 2018. 176, 179

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. The Kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 8, 35, 130

Ke, Q., Bennamoun, M., An, S., Sohel, F., and Boussaid, F. A new representation of skeleton sequences for 3D action recognition. In *CVPR*, 2017. 125, 139

Keskin, C., Kıraç, F., Kara, Y., and Akarun, L. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*, 2012. 172, 175

Kinect. https://en.wikipedia.org/wiki/Kinect. 175

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2014. 198, 202

Kläser, A., Marszałek, M., and Schmid, C. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008. 33

Kong, Y., Ding, Z., Li, J., and Fu, Y. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 26(6):3028–3037, 2017. 123, 125

Kong, Y. and Fu, Y. Human action recognition and prediction: A survey. *CoRR*, abs/1806.11230, 2018. 124

Kostrikov, I. and Gall, J. Depth sweep regression forests for estimating 3D human pose from images. In *BMVC*, 2014. 64, 87

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 34, 63, 97, 98, 99, 163

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 2, 24, 105, 151, 153, 155

Kuehne, H., Arslan, A. B., and Serre, T. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 152, 154

Lan, Z.-Z., Lin, M., Li, X., Hauptmann, A. G., and Raj, B. Beyond Gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, 2015. 116

Laptev, I. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 33

Laptev, I. *Modeling and visual recognition of human actions and interactions*. Habilitation à diriger des recherches en mathématiques et en informatique, Ecole normale supérieure, Paris, France, 2013. 4

Laptev, I., Marszałek, M., Schmid, C., and Rozenfeld, B. Learning realistic human actions from movies. In *CVPR*, 2008. 33, 34, 97, 151

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. Unite the people: Closing the loop between 3D and 2D human representations. In *CVPR*, 2017. 2, 25, 26, 64, 65, 72, 74, 75, 80, 81, 85, 87

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 63, 99

Lee, H.-J. and Chen, Z. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148 – 168, 1985. 18, 20

Lenz, I., Lee, H., and Saxena, A. Deep learning for detecting robotic grasps. In *The International Journal of Robotics Research*, 2015. 183

Leroy, V., Franco, J.-S., and Boyer, E. Multi-view dynamic shape refinement using local temporal integration. In *ICCV*, 2017. 62

Lewiner, T., Lopes, H., Vieira, A. W., and Tavares, G. Efficient implementation of marching cubes cases with topological guarantees. *Journal of Graphics Tools*, 8(2): 1–15, 2003. 72, 73, 87

Li, S. and Chan, A. B. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014. 22

Lin, J., Wu, Y., and Huang, T. S. Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion*, 2000. 178

Lin, T.-Y., 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. In *ECCV*, 2014. 20

Liu, F., Shen, C., and Lin, G. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, 2015. 25, 46

Liu, J., Wang, G., Hu, P., Duan, L., and Kot, A. C. Global context-aware attention LSTM networks for 3D action recognition. In *CVPR*, 2017a. 125, 139

Liu, J., Luo, J., and Shah, M. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. 151, 153

Liu, J., Kuipers, B., and Savarese, S. Recognizing human actions by attributes. In *CVPR*, 2011a. 34

Liu, J., Shah, M., Kuipers, B., and Savarese, S. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011b. 122

Liu, J., Shahroudy, A., Xu, D., and Wang, G. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *ECCV*, 2016. 125, 139

Liu, M. and Yuan, J. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018. 125, 139

Liu, M., Liu, H., and Chen, C. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recogn.*, 68(C):346–362, 2017b. 123, 125, 139

Lomonaco, V. and Maltoni, D. Core50: a new dataset and benchmark for continuous object recognition. In *Proceedings of the 1st Annual Conference on Robot Learning*, Proceedings of Machine Learning Research, 2017. 192, 204

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 24

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. SMPL: A skinned multi-person linear model. In *SIGGRAPH Asia*, 2015. 7, 12, 26, 27, 38, 42, 63, 65, 70, 72, 92, 177, 184

Loper, M. M., Mahmood, N., and Black, M. J. MoSh: Motion and shape capture from sparse markers. In *SIGGRAPH Asia*, 2014. 12, 38, 43, 45, 62

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 33

LTC project page. http://www.di.ens.fr/willow/research/ltc/. 99, 118

Lucchi, A., Li, Y., Boix, X., Smith, K., and Fua, P. Are spatial and global constraints really necessary for segmentation? In *ICCV*, 2011. 24

Luo, Z., Hsieh, J.-T., Jiang, L., Niebles, J. C., and Fei-Fei, L. Graph distillation for action detection with privileged information. In *ECCV*, 2018. 125, 139, 143

Luvizon, D. C., Picard, D., and Tabia, H. 2D/3D pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018. 66, 125, 139, 143

MacCormick, J. and Isard, M. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV*, 2000. 172

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., and Goldberg, K. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. 2017. 183

Maire, M., Yu, S. X., and Perona, P. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011. 24

Malik, J., Elhayek, A., Nunnari, F., Varanasi, K., Tamaddon, K., Héloir, A., and Stricker, D. DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *3DV*, 2018. 173, 174, 175

Mansimov, E., Srivastava, N., and Salakhutdinov, R. Initialization strategies of spatio-temporal convolutional neural networks. *CoRR*, abs/1503.07274, 2015. 35

Marin, J., Vazquez, D., Geronimo, D., and Lopez, A. M. Learning appearance in virtual scenarios for pedestrian detection. In *CVPR*, 2010. 8, 40

Marr, D., Nishihara, H. K., and Brenner, S. Representation and recognition of the spatial organization of three-dimensional shapes. In *Royal Society of London B*, 1978. 18, 19, 31

Marszałek, M., Laptev, I., and Schmid, C. Actions in context. In *CVPR*, 2009. 154

Martinez, J., Hossain, R., Romero, J., and Little, J. J. A simple yet effective baseline for 3D human pose estimation. In *ICCV*, 2017. 22, 62, 64

Massa, F., Russell, B., and Aubry, M. Deep exemplar 2D-3D detection by adapting from real to rendered views. In *CVPR*, 2016. 127

Maturana, D. and Scherer, S. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IROS*, 2015. 63, 176

Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., and Theobalt, C. VNect: Real-time 3D human pose estimation with a single RGB camera. *SIGGRAPH*, 2017. 22

Miller, A. T. and Allen, P. K. Graspit! A versatile simulator for robotic grasping. *Robotics Automation Magazine, IEEE*, 11:110 – 122, 2004. 174, 183, 184, 202, 203

Miller, G. A. English verbs of motion: a case study in semantics and lexical memory. In *Coding Processes and Human Memory*, 1972. 29, 30

Min, P. Binvox. http://www.patrickmin.com/binvox. 67

Moeslund, T. B., Hilton, A., Krger, V., and Sigal, L. *Visual Analysis of Humans: Looking at People*. Springer Publishing Company, Incorporated, 2013. 19

Möller, T. and Trumbore, B. Fast, minimum storage ray-triangle intersection. *J. Graph. Tools*, 1997. 181

Moreno-Noguer, F. 3D human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017. 22

Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *ICCV*, 2017. 175

Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018. 173, 174, 175, 196, 197

Nagel, H.-H. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6(2):59 – 74, 1988. 10

Neumann, B. and Novak, H.-J. Event models for recognition and natural language description of events in real-world image sequences. In *IJCAI*, 1983. 29

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 19, 20, 21, 24, 38, 46, 47, 62, 63, 69, 71, 84, 92

Newell, A., Huang, Z., and Deng, J. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, 2017. 21

Ng, J. Y., Choi, J., Neumann, J., and Davis, L. S. ActionFlowNet: Learning motion representation for action recognition. In *WACV*, 2018. 35

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., and Toderici, G. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 34, 116

Niebles, J. C., Wang, H., and Fei-Fei, L. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008. 97

Niyogi, S. A. and Adelson, E. H. Analyzing gait with spatiotemporal surfaces. In *Motion of Non-Rigid and Articulated Obgects Workshop*, 1994. 32, 33

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, 2006. 73

Nooruddin, F. S. and Turk, G. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):191–205, 2003. 67

ObMan project page. http://www.di.ens.fr/willow/research/obman/. 174

Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.-C., Lee, J. T., Mukherjee, S., Aggarwal, J., Lee, H., Davis, L., et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011. 152

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*, 2011a. 172

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, 2011b. 176

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. Tracking the articulated motion of two strongly interacting hands. In *CVPR*, 2012. 176

Okada, R. and Soatto, S. Relevant feature selection for human pose estimation and localization in cluttered images. In *ECCV*, 2008. 8, 40

Oliveira, G., Valada, A., Bollen, C., Burgard, W., and Brox, T. Deep learning for human part discovery in images. In *ICRA*, 2016. 25, 46, 48, 51, 52

Omran, M., Lassner, C., Pons-Moll, G., Gehler, P. V., and Schiele, B. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, 2018. 28

O'Rourke, J. and Badler, N. I. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(6):522–536, 1980. 18, 19

Panteleris, P., Oikonomidis, I., and Argyros, A. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*, 2018. 173, 175

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A., Tzionas, D., and Black, M. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 26, 27, 28

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017. 22, 62, 64, 70

Pavlakos, G., Zhou, X., and Daniilidis, K. Ordinal depth supervision for 3D human pose estimation. In *CVPR*, 2018a. 23

Pavlakos, G., Zhu, L., Zhou, X., and Daniilidis, K. Learning to estimate 3D human pose and shape from a single color image. In *CVPR*, 2018b. 23, 28, 177

Peng, X., Sun, B., Ali, K., and Saenko, K. Learning deep object detectors from 3D models. In *ICCV*, 2015. 39

Pentland, A. and Horowitz, B. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:730–742, 1991. 31, 32

Perronnin, F., Sánchez, J., and Mensink, T. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010. 33, 163

Pfister, T. *Advancing Human Pose and Gesture Recognition*. PhD thesis, University of Oxford, 2015. 19

Pham, T., Kyriazis, N., Argyros, A. A., and Kheddar, A. Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 176

Pirsiavash, H. and Ramanan, D. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 154, 155

Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormählen, T., and Schiele, B. Learning people detection models from few training samples. In *CVPR*, 2011. 8, 40

Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T., and Schiele, B. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*, 2012. 8, 40

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., and Schiele, B. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016. 19, 21, 62

Pons-Moll, G., Romero, J., Mahmood, N., and Black, M. J. Dyna: A model of dynamic human shape in motion. In *SIGGRAPH*, 2015. 44

Popa, A., Zanfir, M., and Sminchisescu, C. Deep multitask architecture for integrated 2D and 3D human sensing. In *CVPR*, 2017. 22, 66

PrimeSense. https://en.wikipedia.org/wiki/PrimeSense. 175

Qiu, W. Generating human images and ground truth using computer graphics. Master's thesis, UCLA, 2016. 40

Rad, M., Oberweger, M., and Lepetit, V. Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In *CVPR*, 2018. 127, 134

Rahmani, H., Mian, A., and Shah, M. Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):667–681, 2018. 125

Rahmani, H. and Mian, A. Learning a non-linear knowledge transfer model for cross-view action recognition. In *CVPR*, 2015. 40, 41, 138

Rahmani, H. and Mian, A. 3D action recognition from novel viewpoints. In *CVPR*, 2016. 40, 41

Rahmani, H., Mahmood, A., Huynh, D., and Mian, A. Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2430–2443, 2016. 124

Ramakrishna, V., Kanade, T., and Sheikh, Y. Reconstructing 3D human pose from 2D image landmarks. In *ECCV*, 2012. 22

Ramanan, D. Learning to parse images of articulated bodies. In *NIPS*, 2006. 19

Ramanan, D., Forsyth, D. A., and Zisserman, A. Strike a pose: Tracking people by finding stylized poses. In *CVPR*, 2005. 19

Rehg, J. M. and Kanade, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, pages 35–46, 1994. 172, 175

Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.-P., Schiele, B., and Theobalt, C. EgoCap: Egocentric marker-less motion capture with two fisheye cameras. In *SIGGRAPH Asia*, 2016. 41

Riegler, G., Ulusoy, A. O., Bischof, H., and Geiger, A. OctNetFusion: Learning depth fusion from data. In *3DV*, 2017a. 63

Riegler, G., Ulusoy, A. O., and Geiger, A. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, 2017b. 63

Robinette, K., Blackwell, S., Daanen, H., Boehmer, M., Fleming, S., Brill, T., Hoeferlin, D., and Burnsides, D. Civilian American and European Surface Anthropometry Resource (CAESAR), Final Report, 2002. 43, 184

Rodriguez, M. D., Ahmed, J., and Shah, M. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 151

Rogez, G. and Schmid, C. MoCap-guided data augmentation for 3D pose estimation in the wild. In *NIPS*, 2016. 8, 22, 40, 53, 64, 87

Rogez, G., Khademi, M., Supančič III, J. S., Montiel, J. M. M., and Ramanan, D. 3D hand pose detection in egocentric RGB-D images. In *ECCV Workshop on Consumer Depth Cameras for Computer Vision*, 2014. 176

Rogez, G., Supančič III, J. S., and Ramanan, D. First-person pose recognition using egocentric workspaces. In *CVPR*, 2015a. 176

Rogez, G., Supančič III, J. S., and Ramanan, D. Understanding everyday hands in action from RGB-D images. In *ICCV*, 2015b. 176

Rogez, G., Weinzaepfel, P., and Schmid, C. LCR-Net: Localization-classification-regression for human pose. In *CVPR*, 2017. 22, 62, 64, 86, 87

Rohr, K. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94 – 115, 1994. 31

Rohr, K. Human movement analysis based on explicit motion models. In Shah, M. and Jain, R., editors, *Motion-Based Recognition*, pages 171–198. Springer, Dordrecht, 1997. 31

Rohrbach, A., Rohrbach, M., Qiu, W., Friedrich, A., Pinkal, M., and Schiele, B. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition*, pages 184–195. Springer, 2014. 152, 154

Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. A dataset for movie description. In *CVPR*, 2015. 151, 154, 155

Rohrbach, M., Amin, S., Andriluka, M., and Schiele, B. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012. 152, 154, 155

Romero, J., Kjellström, H., and Kragic, D. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *ICRA*, 2010. 176, 177

Romero, J., Loper, M., and Black, M. J. FlowCap: 2D human pose from optical flow. In *GCPR*, 2015. 40

Romero, J., Tzionas, D., and Black, M. J. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 13, 173, 177, 178, 184

Ronfard, R., Schmid, C., and Triggs, B. Learning to parse pictures of people. In *ECCV*, 2002. 18, 20

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 24

Rozantsev, A., Salzmann, M., and Fua, P. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 127

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 34, 177, 185, 192, 202

Ryoo, M. S. and Aggarwal, J. K. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009. 154

Sadanand, S. and Corso, J. J. Action bank: A high-level representation of activity in video. In *CVPR*, 2012. 34

Sahbani, A., El-Khoury, S., and Bidaud, P. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 2012. 183

Salton, G. and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840. 156

Sapp, B. and Taskar, B. MODEC: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013. 20, 38

Schüldt, C., Laptev, I., and Caputo, B. Recognizing human actions: a local SVM approach. In *ICPR*, 2004. 33, 97, 152, 153

Scovanner, P., Ali, S., and Shah, M. A 3-dimensional SIFT descriptor and its application to action recognition. In *ACM International Conference on Multimedia*, 2007. 33

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., and Levine, S. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018. 127, 137

Shah, M. and Jain, R. *Motion-Based Recognition*. Springer, Dordrecht, 1997. 29

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 2, 122, 123, 124, 125, 133, 139

Shapovalova, N., Fernández, C., Roca, F. X., and Gonzàlez, J. Semantics of human behavior in image sequences. In Salah, A. A. and Gevers, T., editors, *Computer Analysis of Human Behavior*, pages 151–182. Springer London, 2011. 10

Shotton, J., Fitzgibbon, A., , Blake, A., Kipman, A., Finocchio, M., Moore, R., and Sharp, T. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011. 19, 24, 41, 175

Sidenbladh, H. and Black, M. J. Learning image statistics for bayesian tracking. In *ICCV*, 2001. 31

Sidenbladh, H., Black, M. J., and Fleet, D. J. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, 2000. 31

Sigal, L., Balan, A., and Black, M. J. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, 2010. 23

Sigal, L., Balan, A., and Black, M. J. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NIPS*, 2008. 26

Sigurdsson, G. A., Russakovsky, O., Farhadi, A., Laptev, I., and Gupta, A. Much ado about time: Exhaustive annotation of temporal data. *arXiv preprint arXiv:1607.07429*, 2016a. 157, 158

Sigurdsson, G. A., Varol, G., Wang, X., Laptev, I., Farhadi, A., and Gupta, A. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016b. 3, 8, 11, 13

Sigurdsson, G. A., Gupta, A., Schmid, C., Farhadi, A., and Alahari, K. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 127, 128, 137

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 25

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 173, 175

Simoncelli, E. P. and Olshausen, B. A. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001. 159

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. 2015. 163

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 34, 35, 66, 97, 98, 99, 100, 103, 104, 108, 110, 115, 116, 163

Sminchisescu, C., Kanaujia, A., and Metaxas, D. Learning joint top-down and bottom-up processes for 3D visual inference. In *CVPR*, 2006. 40

Sminchisescu, C. and Triggs, B. Kinematic jump processes for monocular 3d human tracking. In *CVPR*, 2003. 31

Song, Y., Goncalves, L., and Perona, P. Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):814–827, 2003. 32

Soomro, K., Roshan Zamir, A., and Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 2, 105, 122, 151, 153, 155

Spurr, A., Song, J., Park, S., and Hilliges, O. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 173

Sridhar, S., Mueller, F., Zollhoefer, M., Casas, D., Oulasvirta, A., and Theobalt, C. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *ECCV*, 2016. 173, 176

Stenger, B., Mendonça, P. R., and Cipolla, R. Model-based 3D tracking of an articulated hand. In *CVPR*, 2001. 172

Su, H., Qi, C. R., Li, Y., and Guibas, L. J. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *ICCV*, 2015. 39

Su, H., Fan, H., and Guibas, L. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017a. 63, 176

Su, H., Qi, C., Mo, K., and Guibas, L. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017b. 63

Subramaniam, A., Chatterjee, M., and Mittal, A. Deep neural networks with inexact matching for person re-identification. In *NIPS*, 2016. 127

Sun, L., Jia, K., Yeung, D.-Y., and Shi, B. E. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015a. 35

Sun, X., Wei, Y., Liang, S., Tang, X., and Sun, J. Cascaded hand pose regression. 2015b. 196, 197

SURREAL project page. http://www.di.ens.fr/willow/research/surreal/. 39, 41

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 97, 99

Tan, V., Budvytis, I., and Cipolla, R. Indirect deep structured learning for 3D human body shape and pose prediction. In *BMVC*, 2017. 28, 63, 65, 74, 75, 80, 81

Tang, D., Yu, T.-H., and Kim, T.-K. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013. 172

Tatarchenko, M., Dosovitskiy, A., and Brox, T. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *ICCV*, 2017. 63

Taylor, G. W., Fergus, R., LeCun, Y., and Bregler, C. Convolutional learning of spatio-temporal features. In *ECCV*, 2010. 34, 98, 99, 100

Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. Direct prediction of 3D body poses from motion compensated sequences. In *CVPR*, 2016. 22

Tekin, B., Bogo, F., and Pollefeys, M. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019. 177

Tieleman, T. and Hinton, G. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012. 72

Tome, D., Russell, C., and Agapito, L. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *CVPR*, 2017. 22

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014a. 20

Tompson, J., Stein, M., Lecun, Y., and Perlin, K. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169:1–169:10, 2014b. 172

Torabi, A., Pal, C., Larochelle, H., and Courville, A. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 154

Toshev, A. and Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 19, 20, 63

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015. 34, 98, 99, 100, 101, 102, 104, 107, 111, 114, 115, 116, 120, 128, 164

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 35

Tsoli, A. and Argyros, A. Joint 3D tracking of a deformable object in interaction with a hand. In *ECCV*, 2018. 176, 177

Tuite, K., Snavely, N., Hsiao, D.-y., Tabing, N., and Popovic, Z. PhotoCity: training experts at large-scale image acquisition through a competitive game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011. 154

Tulsiani, S., Zhou, T., Efros, A. A., and Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 68

Tung, H., Tung, H., Yumer, E., and Fragkiadaki, K. Self-supervised learning of motion capture. In *NIPS*, 2017. 28, 63, 65, 79

Tversky, B., Morrison, J., and Zacks, J. On bodies and events. In Meltzoff, A. and Prinz, W., editors, *The Imitative Mind*. Cambridge University Press, 2002. 97

Tzionas, D. and Gall, J. 3d object reconstruction from hand-object interactions. In *ICCV*, 2015. 176

Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., and Gall, J. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016. 173, 176, 177, 186, 187

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008. 159

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. Learning from synthetic humans. In *CVPR*, 2017. 8, 11, 12, 25, 64, 66, 69, 70, 71, 72, 74, 84, 184

Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., and Schmid, C. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018a. 11

Varol, G., Laptev, I., and Schmid, C. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (6):1510–1517, 2018b. 11, 12, 35, 128

Varol, G., Laptev, I., and Schmid, C. On view-independent video representations for action recognition. *Work in progress*, 2019. 11

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. Sequence to sequence-video to text. In *ICCV*, 2015. 168

Vo, N. N. and Hays, J. Localizing and orienting street views using overhead imagery. In *ECCV*, 2016. 127

von Marcard, T., Rosenhahn, B., Black, M., and Pons-Moll, G. Sparse inertial poser: Automatic 3D human pose estimation from sparse IMUs. In *Eurographics*, 2017. 62

Wachter, S. and Nagel, H. . Tracking of persons in monocular image sequences. In *IEEE Nonrigid and Articulated Motion Workshop*, 1997. 31

Wang, D., Ouyang, W., Li, W., and Xu, D. Dividing and aggregating network for multi-view action recognition. In *ECCV*, 2018a. 123, 125, 138, 139, 141, 143

Wang, H. and Schmid, C. Action recognition with improved trajectories. In *ICCV*, 2013. 33, 97, 98, 114, 115, 116, 162, 164

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013a. 33, 34

Wang, J., Nie, X., Xia, Y., Wu, Y., and Zhu, S. Cross-view action modeling, learning, and recognition. In *CVPR*, 2014. 124, 131

Wang, L., Qiao, Y., and Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015a. 35, 99, 116

Wang, L., Qiao, Y., and Tang, X. Motionlets: Mid-level 3D parts for human motion recognition. In *CVPR*, 2013b. 34

Wang, L., Xiong, Y., Wang, Z., and Qiao, Y. Towards good practices for very deep two-stream convnets. In *arXiv:1507.02159*, 2015b. 34, 35, 100, 104

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Val Gool, L. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016a. 34, 35, 138

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *ECCV*, 2018b. 176, 179, 199

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *SIGGRAPH*, 2017. 63

Wang, X., Farhadi, A., and Gupta, A. Actions ~ transformations. In *CVPR*, 2016b. 116

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *CVPR*, 2018c. 35

Wang, Y., Min, J., Zhang, J., Liu, Y., Xu, F., Dai, Q., and Chai, J. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)*, 32(4):43:1–43:14, 2013c. 176

Wang, Y.-X. and Hebert, M. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 127

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. Convolutional pose machines. In *CVPR*, 2016. 19, 20, 21, 38, 62, 63

Weinland, D., Boyer, E., and Ronfard, R. Action recognition from arbitrary views using 3D exemplars. In *ICCV*, 2007. 124

Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, W. T., and Tenenbaum, J. B. MarrNet: 3D shape reconstruction via 2.5D sketches. In *NIPS*, 2017. 176

Wu, Y., Lin, J. Y., and Huang, T. S. Capturing natural hand articulation. In *ICCV*, 2001. 172

Yacoob, Y. and Black, M. J. Parameterized modeling and recognition of activities. In *ICCV*, 1998. 32

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *NIPS*, 2016. 63, 65, 68

Yang, J., Franco, J.-S., Hétroy-Wheeler, F., and Wuhrer, S. Estimation of human body shape in motion with wide clothing. In *ECCV*, 2016. 62

Yang, Y. and Ramanan, D. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 19

Yang, Y. and Ramanan, D. Articulated human detection with flexible mixtures of parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2878–2890, 2013. 19

Yasin, H., Iqbal, U., Kruger, B., Weber, A., and Gall, J. A dual-source approach for 3D pose estimation from a single image. In *CVPR*, 2016. 22, 53, 64, 87

Yi, K. M., Trulls Fortuny, E., Lepetit, V., and Fua, P. LIFT: Learned invariant feature transform. In *ECCV*, 2016. 127

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*, 2015. 45, 185

Yumer, M. E. and Mitra, N. J. Learning semantic deformation flows with 3D convolutional networks. In *ECCV*, 2016. 63

Zagoruyko, S. and Komodakis, N. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015. 127

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 118

Zelnik-Manor, L. and Irani, M. Event-based analysis of video. In *CVPR*, 2001. 32, 33

Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. Real-time action recognition with enhanced motion vector CNNs. In *CVPR*, 2016. 100

Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., and Yang, Q. 3D hand pose tracking and estimation using stereo matching. *arXiv:1610.07214*, 2016. 194, 195, 196

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *ICCV*, 2017. 123, 125

Zheng, J. and Jiang, Z. Learning view-invariant sparse representations for cross-view action recognition. In *ICCV*, 2013. 125

Zheng, J., Jiang, Z., and Chellappa, R. Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing*, 25(6):2542–2556, 2016. 122, 125

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 97, 99, 151

Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. Sparseness meets deepness: 3D human pose estimation from monocular video. In *CVPR*, 2016a. 22, 40

Zhou, X., Sun, X., Zhang, W., Liang, S., and Wei, Y. Deep kinematic pose regression. In *ECCV Workshop on Geometry Meets Deep Learning*, 2016b. 22

Zhou, X., Huang, Q., Sun, X., Xue, X., and Wei, Y. Towards 3D human pose estimation in the wild: A weakly-supervised approach. In *ICCV*, 2017. 22, 25, 62, 64

Zhu, J., Wang, B., Yang, X., Zhang, W., and Tu, Z. Action recognition with actons. In *ICCV*, 2013. 34

Zhu, R., Kiani, H., Wang, C., and Lucey, S. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *ICCV*, 2017. 68

Zimmermann, C. and Brox, T. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. 173, 174, 175, 186, 196, 197

Zipf, G. K. *The psycho-biology of language*. Houghton, Mifflin, 1935. 159

Zitnick, C. and Parikh, D. Bringing semantics into focus using visual abstraction. In *CVPR*, 2013. 155

Zolfaghari, M., Oliveira, G. L., Sedaghat, N., and Brox, T. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*, 2017. 25, 35, 125, 139, 143

RÉSUMÉ

Le contenu visuel se concentre souvent sur les humains. L'analyse automatique des humains à partir de données visuelles revêt donc une grande importance pour de nombreuses applications. Le but de cette thèse est d'apprendre des représentations visuelles pour l'analyse des humains. Un accent particulier est mis sur deux domaines étroitement liés de la vision artificielle: l'analyse du corps humain et la reconnaissance des actions. En résumé, nos contributions sont les suivantes: (i) nous générons des données synthétiques photoréalistes de personnes permettant l'entraînement de CNNs pour l'analyse du corps humain, (ii) nous proposons une architecture multitâche permettant d'obtenir une représentation volumétrique du corps à partir d'une seule image, (iii) nous étudions les avantages des convolutions temporelles à long terme pour la reconnaissance de l'action humaine à l'aide de CNNs 3D, (iv) nous incorporons une fonction de coût de similarité des vidéos multi-vues pour concevoir des représentations invariantes au changement de vue.

MOTS CLÉS

Vision artificielle, analyse du corps humain, reconnaissance de l'action humaine, réseaux de neurones convolutionnels, apprentissage de la représentation.

ABSTRACT

The focus of visual content is often people. Automatic analysis of people from visual data is therefore of great importance for numerous applications in content search, autonomous driving, surveillance, health care, and entertainment. The goal of this thesis is to learn visual representations for human understanding. Particular emphasis is given to two closely related areas of computer vision: human body analysis and human action recognition. In summary, our contributions are the following: (i) we generate photo-realistic synthetic data for people that allows training CNNs for human body analysis, (ii) we propose a multi-task architecture to recover a volumetric body shape from a single image, (iii) we study the benefits of long-term temporal convolutions for human action recognition using 3D CNNs, (iv) we incorporate similarity training in multi-view videos to design view-independent representations for action recognition.

KEYWORDS

Computer vision, human body analysis, human action recognition, convolutional neural networks, representation learning.