Towards Accurate Multi-Person Pose Estimation in the Wild

A dissertation submitted towards the degree of
Doctor of Engineering (Dr.-Ing.)
of the Faculty of Mathematics and Computer Science
of Saarland University

by
Eldar Insafutdinov, M.Sc.

Saarbrücken, 2020

Date of defense	16th of December, 2020
Dean of the faculty	Univ.-Prof. Dr. Thomas Schuster

Examination Committee

Chair	Prof. Dr. Antonio Krüger
Reviewer, advisor	Prof. Dr. Bernt Schiele
Reviewer	Prof. Dr. Bodo Rosenhahn
Reviewer	Prof. Dr. Jia Deng
Academic assistant	Dr. Mohamed Elgharib

ABSTRACT

In this thesis we are concerned with the problem of articulated human pose estimation and pose tracking in images and video sequences. Human pose estimation is a task of localising major joints of a human skeleton in natural images and is one of the most important visual recognition tasks in the scenes containing humans with numerous applications in robotics, virtual and augmented reality, gaming and healthcare among others. Articulated human pose tracking requires tracking multiple persons in the video sequence while simultaneously estimating full body poses. This task is important for analysing surveillance footage, activity recognition, sports analytics, etc. Most of the prior work focused on the pose estimation of single pre-localised humans whereas here we address a case with multiple people in real world images which entails several challenges such as person-person overlaps in highly crowded scenes, unknown number of people or people entering and leaving video sequences.

The first contribution is a multi-person pose estimation algorithm based on the bottom-up detection-by-grouping paradigm. Unlike the widespread top-down approaches our method detects body joints and pairwise relations between them in a single forward pass of a convolutional neural network. Multi-person parsing is performed by optimizing a joint objective based on a multicut graph partitioning framework. Secondly, we extend our pose estimation approach to articulated multi-person pose tracking in videos. Our approach performs multi-target tracking and pose estimation in a holistic manner by optimising a single objective. We further simplify and refine the formulation which allows us to reach close to the real-time performance. Thirdly, we propose a large scale dataset and a benchmark for articulated multi-person tracking. It is the first dataset of video sequences comprising complex multi-person scenes and fully annotated tracks with 2D keypoints. Our fourth contribution is a method for estimating 3D body pose using on-body wearable cameras. Our approach uses a pair of downward facing, head-mounted cameras and captures an entire body. This egocentric approach is free of limitations of traditional setups with external cameras and can estimate body poses in very crowded environments. Our final contribution goes beyond human pose estimation and is in the field of deep learning of 3D object shapes. In particular, we address the case of reconstructing 3D objects from weak supervision. Our approach represents objects as 3D point clouds and is able to learn them with 2D supervision only and without requiring camera pose information at training time. We design a differentiable renderer of point clouds as well as a novel loss formulation for dealing with camera pose ambiguity.

ZUSAMMENFASSUNG

In dieser Arbeit behandeln wir das Problem der Schätzung und Verfolgung artikulierter menschlicher Posen in Bildern und Video-Sequenzen. Die Schätzung menschlicher Posen besteht darin die Hauptgelenke des menschlichen Skeletts in natürlichen Bildern zu lokalisieren und ist eine der wichtigsten Aufgaben der visuellen Erkennung in Szenen, die Menschen beinhalten. Sie hat zahlreiche Anwendungen in der Robotik, virtueller und erweiterter Realität, in Videospielen, in der Medizin und weiteren Bereichen. Die Verfolgung artikulierter menschlicher Posen erfordert die Verfolgung mehrerer Personen in einer Videosequenz bei gleichzeitiger Schätzung vollständiger Körperhaltungen. Diese Aufgabe ist besonders wichtig für die Analyse von Video-Überwachungsaufnahmen, Aktivitätenerkennung, digitale Sportanalyse etc. Die meisten vorherigen Arbeiten sind auf die Schätzung einzelner Posen vorlokalisierter Menschen fokussiert, wohingegen wir den Fall mehrerer Personen in natürlichen Aufnahmen betrachten. Dies bringt einige Herausforderungen mit sich, wie die Überlappung verschiedener Personen in dicht gedrängten Szenen, eine unbekannte Anzahl an Personen oder Personen die das Sichtfeld der Video-Sequenz verlassen oder betreten.

Der erste Beitrag ist ein Algorithmus zur Schätzung der Posen mehrerer Personen, welcher auf dem Paradigma der Erkennung durch Gruppierung aufbaut. Im Gegensatz zu den verbreiteten Verfeinerungs-Ansätzen erkennt unsere Methode Körpergelenke and paarweise Beziehungen zwischen ihnen in einer einzelnen Vorwärtsrechnung eines faltenden neuronalen Netzwerkes. Die Gliederung in mehrere Personen erfolgt durch Optimierung einer gemeinsamen Zielfunktion, die auf dem Mehrfachschnitt-Problem in der Graphenzerlegung basiert. Zweitens erweitern wir unseren Ansatz zur Posen-Bestimmung auf das Verfolgen mehrerer Personen und deren Artikulation in Videos. Unser Ansatz führt eine Verfolgung mehrerer Ziele und die Schätzung der zugehörigen Posen in ganzheitlicher Weise durch, indem eine einzelne Zielfunktion optimiert wird. Desweiteren vereinfachen und verfeinern wir die Formulierung, was unsere Methode nah an Echtzeit-Leistung bringt. Drittens schlagen wir einen großen Datensatz und einen Bewertungsmaßstab für die Verfolgung mehrerer artikulierter Personen vor. Dies ist der erste Datensatz der Video-Sequenzen von komplexen Szenen mit mehreren Personen beinhaltet und deren Spuren komplett mit zwei-dimensionalen Markierungen der Schlüsselpunkte versehen sind. Unser vierter Beitrag ist eine Methode zur Schätzung von drei-dimensionalen Körperhaltungen mittels am Körper tragbarer Kameras. Unser Ansatz verwendet ein Paar nach unten gerichteter, am Kopf befestigter Kameras und erfasst den gesamten Körper. Dieser egozentrische Ansatz ist frei von jeglichen Limitierungen traditioneller Konfigurationen mit externen Kameras und kann Körperhaltungen in sehr dicht gedrängten Umgebungen bestimmen. Unser letzter Beitrag geht über die Schätzung menschlicher Posen hinaus in den Bereich des tiefen Lernens der Gestalt von drei-dimensionalen Objekten. Insbesondere befassen wir uns mit dem Fall drei-dimensionale Objekte unter schwacher Überwachung zu rekonstruieren. Unser Ansatz repräsentiert Objekte als drei-dimensionale Punktwolken and ist im Stande diese nur mittels zwei-dimensionaler Überwachung und ohne Informationen über die Kamera-Ausrichtung zur Trainingszeit zu lernen. Wir entwerfen einen differenzierbaren Renderer für Punktwolken sowie eine neue Formulierung um mit uneindeutigen Kamera-Ausrichtungen umzugehen.

ACKNOWLEDGEMENTS

First and foremost, I owe a sincere gratitude to my supervisor Bernt Schiele. He took me as a master student and guided me through the entire Ph.D. journey. It is incredible to think how many successful careers Bernt helped to launch and I am eternally grateful to him for his mentorship. These acknowledgments would also not have been possible without Leonid Pishchulin who was my first supervisor at MPI and introduced me to the computer vision research. I am extremely fortunate to have worked with Misha Andriluka who was my second Ph.D. advisor and whose relentless energy and enthusiasm have kept the voyage in high spirit. I also thank Alexey Dosovitsky for hosting me during my internship at Intel and sharing with me a his expertise on training neural nets. I would like to thank my thesis examiners Bodo Rosenhahn and Jia Deng for reading and evaluating my thesis, as well as Mohamed Elgharib and Antonio Krüger for serving on my Ph.D. committee.

During the Ph.D. I have been extremely fortunate to work with many wonderful people on a number of exciting projects: Bjoern Andres, Gerard Pons-Moll, Helge Rhodin, Christian Richardt and Umar Iqbal among others. The time I spent at MPI would not have been so great without the fantastic colleagues who over time also became friends. A big shout-out to Evgeny Levinkov, Mohamed Omran, Jan-Hendrik Lange, Anna Khoreva, Mateusz Malinowski, Siyu Tang, Bojan Pepik, Jan Hosang, Maxim Lapin, Anna Rohrbach, Julian Steil, Philipp Müller, Hosnieh Sattar, Qianru Sun, Rakshith Shetty, Apratim Bhattacharyya, Tribhuvanesh Orekondy, Verica Lazova, Bharat Lal Bhatnagar, Paul Swoboda, David Stutz, Moritz Böhle, Aymen Mir, Xiaohan Zhang and Anna Kukleva: it has been a tremendous pleasure knowing you and interacting with you over the years. A special thanks to Connie Balzert for making my stay at MPI even more enjoyable, for the afternoon chats and for the invaluable practice sessions of conversational German. The Ph.D. allowed me to meet some amazing people outside of the department. I spent many memorably moments Mira Slavcheva, Yani Ioannou, Pavel Tokmakov, Martin Danelljan, Jack Valmadre and Artsiom Sanakoyeu during the computer vision conferences I was lucky to attend.

I would like to thank my friends whom I met during my 7-year stay in Saarbrücken for all the precious moments we spent together. I am so lucky to have met Dushyant Mehta during our master's studies and our friendship persisted as we both entered the Ph.D. His support and encouragement have been instrumental during the final stages of my Ph.D. including his offer to proofread large portions of the dissertation. A big thanks to Nikita Vedeneev who has been a constant source of inspiration, I learned a lot from you! I would like acknowledge Varvara Obolonchykova, Tanya Dembelova, Konstantin Poddubnyy, Kailash Budhathoki, Baris Cakar, Ayush Tewari, Philipp Flotho and Junaid Ali who have been amazing friends.

I would like to thank my girlfriend Marina Ermashova who was supporting me unconditionally over the past two years. I am blessed to have a sister, Elmira, who has been regularly sharing wisdom with me through the times of uncertainty. Finally, I want to thank my parents, Aksan and Anna, who have made all of this possible in the first place, for their love and encouragement during the years of study and work.

CONTENTS

1 INTRODUCTION

Humans represent one of the most important categories in visual recognition. Enabling understanding of humans in natural and human constructed surroundings, together with an understanding of the structure and nature of objects in the surroundings, through the use of images and videos is crucial for a wide range of applications. These range from robotics to entertainment, health care to sport industry, as well as social sciences which study how humans communicate with each other (Joo, 2019). For example, consider the task of designing an intelligent robot whose purpose is to assist humans in their daily activities. Such a robot must be endowed with a visual system able to sense and understand humans in motion in order to enable seamless human-robot interaction. More broadly, automatic recognition of articulated humans allows for new possibilities in human computer interaction. Furthermore, the recent decade has seen a proliferation of social network services built around photo- and video-sharing, with humans always at the center of the story.

Human pose estimation is a family of computer vision problems and algorithms that can facilitate many of the applications in the aforementioned domains. This thesis addresses 2D human pose estimation from color image or video: given an image or a video containing one or more human subjects the algorithm is required to identify pixel locations corresponding to major skeletal joints of the human body (see Figure 1.1). Human pose estimation is a challenging task due to complex articulations, varying camera viewpoints, clothing, differences in scale, illumination, complex backgrounds, self-occlusion, etc. An example in Figure 1.1 demonstrates some of those challenges: low contrast on the body, cluttered background, significant articulation and self-occlusion (right hip joint is behind the left thigh).

Much of the work in the last three decades focused on estimating poses of single, pre-localized humans (see Figure 1.2 (a)). And while this problem setting had been a driver for a remarkable progress, it does not fully represent the diversity of real-world imagery. Indeed, most photographs and videos contain multiple people of interest which entails unique challenges: a-priory unknown number of people in the scene, partial visibility of some people and person-to-person occlusions as shown in Figure 1.2 (b). There is a clear need for an algorithm that can perform person detection, articulated pose estimation as well as tracking of multiple people in video sequences. These computer vision problems had previously been addressed in isolation without leveraging complementary strengths of different approaches. For example, in case of inter-personal occlusions and overlap, knowing the pose of the occluding person can greatly constrain the search space for the joint locations of the occluded person. Or in the video setting the hard-to-detect body parts can be resolved by propagating confidences from neighbouring frames.

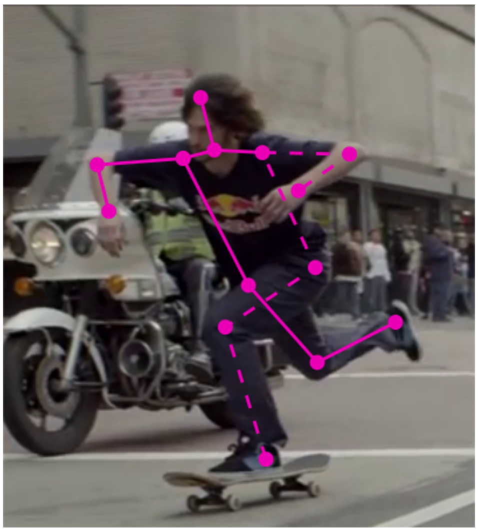
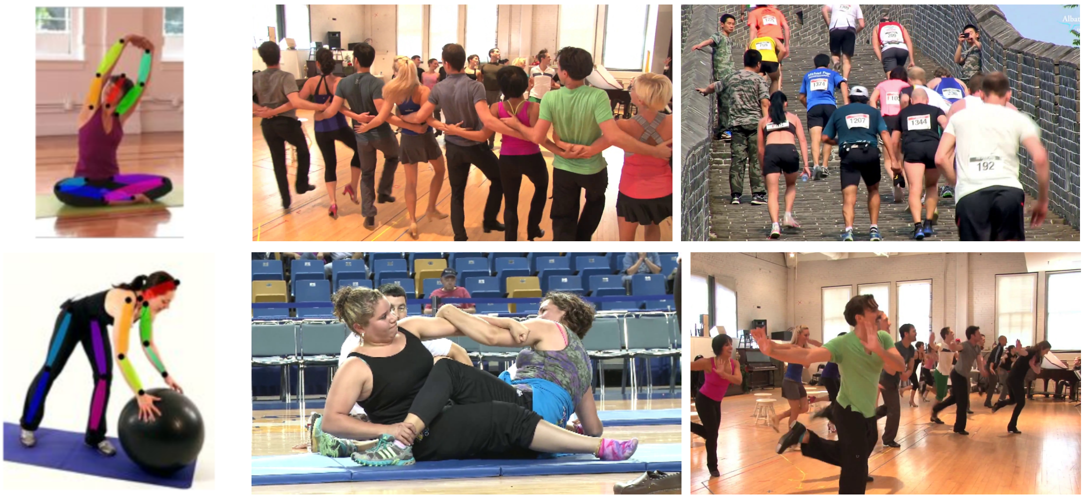


Figure 1.1: An example demonstrating the problem of 2D human pose estimation. An algorithm must infer the locations of the major joints of the human body.



(a) (b)

Figure 1.2: (a) Single-person pose estimation with isolated, pre-localized humans. Prior work addressed largely this setting. (b) Challenging real-world scenes depicting multiple people, interacting with each other in complicated arrangements with a significant degree of overlap. Human detection and pose estimation in such challenging scenarios is the focus of this thesis.

In this thesis we will discuss computational approaches for multi-person pose estimation and tracking in challenging multi-person scenes. Our models build on a framework that jointly estimates poses of all people present in the image. Its formulation falls into a bottom-up detection-by-grouping paradigm and can automatically infer the number of people, resolve ambiguous part associations as well as suppress occluded parts. The methods developed in this thesis innovate on several fronts compared to the state of the art. Firstly, this thesis proposes new and improved body part detectors that significantly boost keypoint localization accuracy. Secondly, we introduce novel image-conditioned pairwise terms used for joint-to-joint association in the bottom-up grouping which allow for a significantly more efficient inference and improved accuracy. Thirdly, we propose several improvements to the formulation which for the first time enable real-time application of multi-person pose estimation. Fourthly, we take advantage of our graph-based formulation and show that it can be naturally extended to multi-person pose tracking in video sequences.

The algorithmic advances in computer vision would not have been possible without the availability of large-scale visual datasets. The 2012 breakthrough in image classification achieved by a deep convolutional network architecture of Krizhevsky *et al.* (2012) was enabled through training on a dataset of Deng *et al.* (2009) consisting of 1M images annotated with class labels. Similarly, datasets with body landmark annotations of increasing size (Johnson and Everingham (2010); Sapp and Taskar (2013); Andriluka *et al.* (2014)) fueled powerful body part detectors based on Convolutional Neural Networks (CNNs) allowing for a significant leap in pose estimation accuracy compared to previous non-deep approaches. Building on our work on multi-person pose tracking in videos this thesis introduces a new benchmark dataset called PoseTrack that sets novel challenges and promotes further research on this task. It is the first dataset with both densely annotated tracks of multiple people and their poses in challenging, crowded scenes.

2D pose estimation is a precursor to the more general problem of 3D pose estimation, which has many practical applications such as animating digital avatar with the human motion for the virtual reality scenario which requires accurate estimates of joint locations in 3D. In this thesis we focus on 2D pose estimation which presents interesting technical challenges on its own. Importantly, it was demonstrated in the literature that complex 3D inference problems can often be decomposed into a 2D image recognition step followed by 3D reasoning. For example, 2D keypoint detection often serves as a useful building block in 3D pose or human shape estimation algorithms. Chapter 6 discusses one such approach that reconstructs a 3D pose of a person using a wearable camera, where 2D keypoint detections are used to constrain 3D joint locations of a skeleton in 3D.

More generally, learning and reasoning about the 3D world from 2D observations is a direction of research that holds significant promise. Collecting three-dimensional data for training of machine learning algorithms requires specialised sensors or multi-view camera setups and such data is less abundant in comparison to monocular images and videos recorded with commodity cameras. In Chapter 7 we demonstrate an algorithm that learns to estimate 3D shape and viewpoint (camera pose) of general object categories from a single view and using only 2D training data. We envision that future developments in this direction will allow to reduce reliance on high quality ground truth data for training 3D reconstruction algorithms.

Learning to represent general object categories is also important for a different reason. Visual scenes rarely depict humans in isolation. On the contrary, we live in human-created world and are surrounded by lots of objects. Systems that are able to reason about such complex scenes need to model human-to-human, object-to-object and human-to-object interactions. Human pose estimation and computational models of object categories are essential components required to build such systems. As reliability of the recognition algorithms is approaching very good levels it allows the research community to start building more ambitious algorithms of holistic scene understanding.

1.1 ORGANIZATION

We will now give descriptions to the chapters of this thesis as well as specify which publications they are based on and note the contributions of individual authors when necessary.

Chapter 2: Related Work Here, we review the literature on human pose estimation, both single- and multi-person settings and video pose estimation and draw connections to the work done in this thesis. We also cover the research that appeared since the publication of our original research including the current state of the art in human pose estimation and articulated tracking.

Chapter 3: Multi-Person Pose Estimation Here, an algorithm for multi-person pose estimation in unconstrained scenes is presented. The previous state-of-the-art on this task (Pishchulin *et al.*, 2016) performs simultaneously person detection and pose estimation by minimizing a joint objective based on an Integer Linear Program. However, it suffers from lower accuracy in crowded scenes as well as prohibitively long inference times. The algorithm presented in Chapter 3 addresses these challenges by (1) an improved deeper architecture for keypoint localization, (2) image-conditioned pairwise terms that help to disambiguate parts of people in close proximity and (3) efficient hierarchical inference which together provide a significant improvement in pose estimation accuracy as well as runtime speed up by several orders of magnitude.

The content of this chapter is based on the ECCV 2016 publication "A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model". Eldar Insafutdinov was the leading author of this publication, proposing its major contributions, and carried out the experiments.

Chapter 4: Articulated Multi-Person Tracking The approach developed in Chapter 3 is capable of performing multi-person pose estimation for a single frame only. In Chapter 4 we tackle a more general problem of articulated tracking of multiple people in video sequences. We propose a unified framework that solves pose estimation and multi-target tracking by minimizing a joint objective. We use a graph partitioning formulation that operates by a bottom-up assembly of part detections within each frame and across time. We additionally propose an end-to-end formulation for associating a body joint to a specific person and embed it in a sparse graph which results in more efficient inference. The experiments demonstrate synergy between tracking and pose estimation, demonstrating higher accuracy than a per-frame pose estimation baseline, especially on hard parts such as limbs and ankles.

The content of this chapter is based on the CVPR 2017 publication "ArtTrack: Articulated Multi-Person Tracking in the Wild". Eldar Insafutdinov was the leading author of this publication and contributed the formulation for multi-person pose tracking, person-conditioned top-down pose estimation model as well as most of the experiments. Siyu Tang contributed the implementation of the pairwise terms for tracking. Mykhaylo Andriluka suggested the idea for the project, substantially improved the tracking terms as well as contributed theoretical formulation for the approach.

Chapter 5: A Benchmark for Human Pose Estimation and Tracking Our work on multi-person pose tracking defined a new computer vision problem. However, our models were trained on single-frame datasets and our annotated validation set was small scale. In order to enable further research we introduce a new large-scale, high-quality benchmark for video-based multi-person pose estimation and articulated tracking. We collect, annotate and release a new dataset that features videos with multiple people labeled with person tracks and articulated pose. Our benchmark is significantly larger and more diverse in terms of data variability and complexity compared to existing pose tracking benchmarks. We provide a public centralized server that runs evaluations on a held-out test set to enable objective comparison of different approaches.

The content of this chapter is based on the CVPR 2018 publication "PoseTrack: A Benchmark for Human Pose Estimation and Tracking". Mykhaylo Andriluka was the lead author of this publication and developed the annotation protocol and the annotation tools. Eldar Insafutdinov managed data annotation as well as evaluated baseline models. Umar Iqbal managed data annotation and provided experimental evaluation with a baseline model. Leonid Pishchulin contributed the evaluation toolkit. Anton Milan developed the website for the benchmark as well as its evaluation server.

Chapter 6: Egocentric Marker-less Motion Capture Human pose estimation framework developed in the previous chapters provides estimates of joint locations in 2D. However, many scenarios require prediction of full 3D skeleton while also respecting various constraints such as joint angle plausibility, bone length symmetry, and smoothness and consistency across frames. Here, we present an approach for 3D motion capture using wearable cameras that combines convolutional 2D detectors developed in Chapter 2 and 3D priors on human body configurations both spatially and temporally. Using on-body camera setup allows robust estimates in presence of cluttered and crowded scenes where the traditional systems with external cameras would struggle.

The content of this chapter is based on the SIGGRAPH Asia 2016 publication "EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras". Helge Rhodin was the leading author of this publication. Eldar Insafutdinov contributed training and evaluation of convolutional part detectors.

Chapter 7: Unsupervised Learning of 3D Object Shape and Camera Pose So far we focused only on the analysis humans, however humans rarely exist in isolation from the surrounding scenes. In this chapter we address learning 3D shape and camera pose of general object categories using weak supervision in the form of unposed 2D object masks. Our system predicts a detailed point clouds from a single image and is supervised by a reprojection back in 2D via a novel differential projection mechanism. The inherent ambiguity in the camera poses is handled by an ensemble of pose predictors trained with the "hindsight" loss. All together this results in very accurate reconstructions of 3D shapes substantially improving over state of the art.

The content of this chapter is based on the NeurIPS 2018 publication "Unsupervised Learning of Shape and Pose with Differentiable Point Clouds". Eldar Insafutdinov was the leading author of this publication.

Part I

Human Pose Estimation and Tracking

RELATED WORK 2

Contents

In this chapter we review the literature on 2D human pose estimation in still images and articulated multi-person tracking in video sequences. We begin with the classical approaches for single person human pose estimation based on pictorial structures and move on to the more recent architectures that utilize convolutional neural networks. We subsequently review the literature on multi-person pose estimation, including top-down and bottom-up methods while drawing connections to the work performed in this thesis. We further study the works related to our proposed approach for articulated multi-person tracking. We cover the publications spanning the earlier approaches for pose estimation in videos of individual people as well as the most recent multi-person pose tracking methods based on advanced deep architectures. Finally, in order to support Chapter 5 on the PoseTrack dataset we review the datasets for human pose estimation in still images and video sequences and provide comparisons to our proposed dataset.

2.1 SINGLE-PERSON POSE ESTIMATION

The pictorial structures models first proposed by Fischler and Elschlager (1973) had been a dominant approach for articulated pose estimation prior to the mass spread of deep learning-based methods in computer vision. Pictorial structures model objects as a collection of parts arranged in a deformable configuration, with the parts captured by local appearance and deformable configuration represented by spring-like connections between pairs of certain parts. More formally, the pictorial structure of an object with N parts is given by a graph $G = (V, E)$, where $V = \{v_i : i \in 1..N\}$ is a set of parts. Matching a pictorial structure model to an image is performed by minimizing the following energy function:

$$L = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j)$$

where l_i denotes the location of the part v_i in the image, $m_i(l_i)$ is the appearance term measuring the cost of placing part v_i at location l_i and $d_ij(l_i, l_j)$ is the pairwise cost that penalises deformations of an object and captures the prior over geometric arrangements of parts. Felzenszwalb and Huttenlocher (2005) propose an efficient algorithm for matching pictorial structures to images. In particular, G is constrained to a tree which is essential for efficient inference. The high computational cost of inference incurred by extremely large pose state space could be mitigated by pruning parts of the state space either based on rough person detection and foreground segmentation (Ferrari *et al.*, 2008) or a course-to-fine scheme (Sapp *et al.*, 2010).

The pictorial structures framework is flexible and allows to incorporate a variety of appearance models and combinations thereof. In particular, several features have been suggested previously including skin and background color (Sapp *et al.*, 2010; Eichner and Ferrari, 2012), Histograms of Oriented Gradients (HOGs) (Buehler *et al.*, 2008; Johnson and Everingham, 2010), part segmentation features (Johnson and Everingham, 2009), segmentation contours (Sapp *et al.*, 2010), pairwise color similarity (Sapp *et al.*, 2010; Tran and Forsyth, 2010) and image motion features (Sapp *et al.*, 2011). Various improvements across various components of the pictorial structures framework further pushed state of the art performance (Andriluka *et al.*, 2009; Yang and Ramanan, 2012; Pishchulin *et al.*, 2013a,b; Kiefel and Gehler, 2014). While the aforementioned works relied on tree-structured graphical models, non-tree models had also been explored in the literature (Bergtholdt *et al.*, 2010; Tran and Forsyth, 2010; Wang and Li, 2013; Dantone *et al.*, 2013). The work of Ramakrishna *et al.* (2014) aim to design stronger detectors by combining the detector output with location-based features.

After the breakthrough results of Krizhevsky *et al.* (2012) for large scale image classification other works followed suit applying Convolutional Neural Networks (CNNs) (LeCun *et al.*, 1998) to the tasks of object localization (Szegedy *et al.*, 2013; Girshick *et al.*, 2014; Sermanet *et al.*, 2014), semantic segmentation (Long *et al.*, 2015; Chen *et al.*, 2017a), depth estimation (Eigen *et al.*, 2014), etc. One of the first methods to apply deep CNNs to the task of human pose estimation is DeepPose proposed by Toshev and Szegedy (2014). The authors directly regress the keypoint coordinates given the input image in a fully-connected fashion. Such formulation, however, suffers from insufficient accuracy especially in the high precision regime, which is attributed to the difficulty of learning a highly non-linear mapping from pixels to coordinates. Toshev and Szegedy (2014) partly mitigate it by learning several cascades of predictors that iteratively refine predictions of coordinates: each subsequent network in the cascade takes as input a crop centered around the prediction of the previous network. Such iterative scheme proved itself useful in subsequent works as will be discussed later. A different way of predicting pose with CNNs is first introduced by Jain *et al.* (2014a) and later refined in (Tompson *et al.*, 2014, 2015). Instead of direct coordinate regression these works pioneer regressing heatmaps representing per-pixel likelihoods of joint locations. This formulation allows to tie network outputs to the pixel evidence in a more direct manner which in turn enables more efficient learning. Additionally, such an approach naturally extends to the multi-person case allowing detectors to fire at multiple locations –

something not possible in the formulation of Toshev and Szegedy (2014). Another important early work by Chen and Yuille (2014) aiming to incorporate convolutional detectors into part-based models. Like in other computer vision problems this completely turned the tide of pose estimation research, with virtually all state-of-the-art methods now exploiting the idea of heatmap regession. The key take away was that years of designing effective image features and highly non-trivial graphical models were superseded by conceptually simpler yet stronger CNN-based detectors.

Wei *et al.* (2016) introduce Convolutional Pose Machines (CPMs). The authors iteratively train a cascade of convolutional parts detectors, each detector taking the scoremap of all parts from the previous stage. This effectively increases the depth of the network which is necessary for spatial reasoning about body configurations. Increasing depth of CNNs allows to increase the size of effective receptive field but unfortunately very deep networks suffer from the problem of vanishing gradients. CPMs mitigate the vanishing gradients problem with intermediate supervision, thus reinforcing the gradients throughout the network. Conceptually similar to this iterative approach are the two other works by Carreira *et al.* (2016) and Newell *et al.* (2016). Carreira *et al.* (2016) describe a cascaded framework where each level predicts an additive refinement to the initial pose prediction. Newell *et al.* (2016) propose a stack of hourglass blocks that performs repeated bottom-up top-down inference, also with the loss attached at each block.

In Chapter 3 we discuss a novel convolutional architecture for body part detection. Our design is motivated by Pishchulin *et al.* (2016), who observe that in the presence of strong detectors explicit spatial reasoning results in diminishing returns because most contextual information can be incorporated directly in the detector. With the recent developments in object detection newer architectures are composed of a large number of layers and the receptive field is large automatically. Our detector is based on deep residual networks He *et al.* (2016) which allows us to train a detector with a large receptive field Wei *et al.* (2016). Our detectors use off-the-shelf ResNet models, their design is simple yet highly effective which is demonstrated by state-of-the-art performance on major single person pose estimation benchmarks.

Subsequently there has been a proliferation of works building on the ideas initially presented in Jain *et al.* (2014b); Tompson *et al.* (2014). Bulat and Tzimiropoulos (2016) propose a two-stage joint detection network where the first stage is trained with cross-entropy loss and the second stage with mean-squared error loss. Belagiannis and Zisserman (2017) propose to use recurrent spatial convolutions which increases effective size of the receptive field while keeping the number of network parameters fixed resulting in more parameter-efficient models. Chu *et al.* (2017) augment the stacked hourglass architecture with multi-resolution attention mechanisms applied at each hourglass. Contrary to the usual practice of generating attention via spatial softmax they use mean-field approximation (Zheng *et al.*, 2015; Krähenbühl and Koltun, 2011) to produce either whole-body part-agnostic or part-specific attention maps, which are eventually applied to the output part-likelihood heatmaps.

Two recent works (Chen *et al.*, 2017b; Chou *et al.*, 2018) propose adversarial formulation where a discriminator distinguishes between ground-truth part-likelihood heatmaps and heatmaps generated by a pose estimation network (stacked hourglass in their case). The discriminator implicitly learns the distribution of human poses and thus encourages the generator to produce more plausible body configurations. Similarly, Wandt and Rosenhahn (2019) train an adversarial critic network in the context of 3D pose estimation, operating on the input representations in joint angles and bone lengths. Fieraru *et al.* (2018) take an alternative approach to address the same issue and train an additional pose refinement network that explicitly learns to fix incorrect predictions. Yang *et al.* (2017) uses feature pyramid networks to enhance detection of joints at multiple scales. Neverova and Kokkinos (2018) refine estimated heatmaps with differentiable geometric voting. Ke *et al.* (2018) propose to train Hourglass networks with multi-scale supervision as well as domain specific data augmentation with synthetic occlusions. Tang *et al.* (2018b) address the efficiency of pose estimation models by proposing quantized densely connected U-Nets (Ronneberger *et al.*, 2015) substantially reducing the number of parameters and network size. Tang *et al.* (2018a) exploit compositional hierarchy of human body by performing repeated bottom-up top-down inference. Sun *et al.* (2019) propose High-Resolution Nets (HRNets), an alternative take on how to perform multi-resolution inference in CNNs: HRNet always maintains a high resolution branch and gradually adds high-to-low resolution subnetworks while maintaining connections between parallel subnetworks of different resolution. Bulat *et al.* (2020) introduce learnable soft gated skip connections for the residual units allowing to significantly improving model efficiency and obtained state-of-the-art pose estimation accuracy using only 1/3 of model parameters of the Stacked Hourglass model.

2.2 MULTI-PERSON POSE ESTIMATION

Earlier work. While single person pose estimation has advanced considerably in the recent decade, its setting remains rather simplified. Single person pose estimation methods require localizing humans in the image and receive as input image regions cropped around the persons of interest. Detecting humans in images and video sequences is a challenge by itself and designing a naive two-stage system that first performs person detection and then pose estimation might not be able to exploit inherent synergies of the two disjoint tasks. In what follows we review the literature on multi-person pose estimation as well as place in its context the work presented in this thesis. Much of the previous work has addressed this problem as sequence of person detection and pose estimation (Eichner and Ferrari, 2010; Ladicky *et al.*, 2013; Chen and Yuille, 2015). Eichner and Ferrari (2010) use a detector for initialization and reasoning across people, but rely on simple geometric body part relationships and only reason about person-person occlusions. Chen and Yuille (2015) focus on single partially occluded people, and handle multi-person scenes akin to Yang and Ramanan (2012).

Bottom-up methods. Pishchulin *et al.* (2016) propose an approach named *DeepCut* that jointly detects and estimates body configurations. It follows a bottom-up paradigm: first, a pool of body part candidates is detected and then the detections are grouped into person instances by minimizing a joint objective responsible for partitioning and labeling. This formulation is based on the multicut problem (Andres, 2015). The main limitation of *DeepCut* is that it relies on simple fixed pairwise terms, which limits the performance and results in prohibitive inference time to fully explore the search space. In Chapter 3 we introduce a multi-person pose estimation algorithm that builds on *DeepCut* and innovates on multiple fronts both in terms of speed and accuracy. The main contributions are image-conditioned pairwise terms derived from a fully convolutional neural network that regresses geometric offsets between different body parts. Compared to fixed pairwise terms our image-conditioned terms aid reducing ambiguities when grouping body part hypotheses into people which results in dramatic improvements of inference times. In Chapter 4 we further improve the model by simplifying the objective function and introducing an efficient solver which, for the first time, enables real-time performance of per-frame multi-person pose estimation.

Notably, Cao *et al.* (2017) introduce OpenPose, a real-time multi-person pose estimation system. Similar to our own work, it detects body joints with a fully convolutional network (Wei *et al.*, 2016) and groups them in a bottom-up fashion by means of image conditioned pairwise terms. Their pairwise terms, named Part Affinity Fields (PAFs), are represented by flow fields of orientated vectors defined over the support region of the limb and are also predicted by a multi-stage fully convolutional network. Grouping of joints is done greedily: people are assembled limb by limb with keypoint association performed via bipartite graph matching. This is in contrast to the formulation employed in this thesis based on an objective function expressed as an integer linear program (ILP). Despite these differences both formulations achieve a similar accuracy on the MPII Human Pose benchmark (Andriluka *et al.*, 2014) as demonstrated in Chapter 4. Hidalgo *et al.* (2019) extend OpenPose to detect facial and hand keypoints with a single network. To this end they introduce a multi-task architecture with a particular emphasis on careful handling inherent differences in scale and resolution between detecting the body keypoints on the one hand and the hand and face keypoints on the other hand.

Newell *et al.* (2017) propose a method for direct end-to-end grouping of joints which they name Associative Embedding. Instead of computing scores for pairs of joints the network directly predicts an identity tag for each body keypoint and is supervised with a triplet-like loss similar to the ones employed in the metric learning literature. Cheng *et al.* (2020) use associative embedding in conjunction with HRNets to achieve state-of-the-art accuracy on MS COCO keypoint detection benchmark (Lin *et al.*, 2014). PersonLab approach introduced in Papandreou *et al.* (2018) is a bottom-up method that uses pairwise terms based on geometric offsets, much like in our own work. The key difference is the use of iterative refinement mechanism for regressing geometric offsets as well as the greedy assembly procedure. Kreiss *et al.* (2019) introduce PiffPaff Composite Fields which are a generalisation of vector fields. In particular, the pairwise offsets are learnt using the Laplace loss popularised in Kendall and Gal (2017). The MultiPoseNet method of Kocabas *et al.* (2018) combines bottom-up and top-down methodologies by refining bottom-up keypoint detections for each person bounding box detection obtained with an integrated RetinaNet-like (Lin *et al.*, 2017) detector.

Top-down methods. Top-down methods have also advanced considerably in the recent years. Papandreou *et al.* (2017) propose a strong multi-pose estimation baseline by combining Faster R-CNN detector (Ren *et al.*, 2015) and heatmap regression. Mask R-CNN (He *et al.*, 2017) is a framework for instance segmentation and human pose estimation that extends Faster R-CNN with a mask prediction and pose estimation branches. State-of-the-art performance in multi-person pose estimation has been held by methods that operate in a top-down fashion by typically combining Faster R-CNN and advanced keypoint detection architecture such as HRNet (Sun *et al.*, 2019). Xia *et al.* (2017) improve pose estimation with part segmentation predictions by joint refinement in a CRF framework. Chen *et al.* (2018) combine a person detector and pose estimator both based on Feature Pyramid Networks (FPNs).

Image-dependent pairwise terms. Computing pairwise scores has driven pose estimation research for decades. In the early days of weakly performing detectors pairwise terms were necessary to represent a prior on human body configurations. Starting with the original work of Felzenszwalb and Huttenlocher (2005) many subsequent methods relied on simple Gaussian pairwise terms that do not depend on the data and therefore referred to as structural priors (Ramanan and Sminchisescu, 2006; Ferrari *et al.*, 2008; Andriluka *et al.*, 2009; Ferrari *et al.*, 2009). The use of such terms for pose inference is suboptimal as they do not take local image evidence into account and only penalize deviation from the expected joint location. Due to the inherent articulation of body parts the expected location of a part with respect to another can only approximately guide the inference.

In this thesis we concentrate on bottom-up multi-person pose estimation where pairwise terms are used to group keypoints into people. In this setting simple pairwise terms can only be sufficient when people are relatively distant from each other, however for closely positioned people more discriminative pairwise costs are essential. Two prior works (Pishchulin *et al.*, 2013a; Chen and Yuille, 2014) introduce image-dependent pairwise terms between connected body parts. While Pishchulin *et al.* (2013a) use an intermediate representation based on poselets our pairwise terms are conditioned directly on the image. Chen and Yuille (2014) cluster relative positions of adjacent joints into $T = 11$ clusters, and assign different labels to the part depending on which cluster it falls to. Subsequently, a CNN is trained to predict this extended set of classes and later an SVM is used to select the maximum scoring joint pair relation.

The image-dependent terms proposed in this thesis are based on a fully convolutional network that predicts relative positions of parts with respect to each other. Then, intuitively, for a pair of body part candidates, if the corresponding offset predictions agree with the actual part locations, there is a strong indication that both parts belong to the same person. To the best of our knowledge, our image-conditioned pairwise were the first to be used for bottom-up multi-person pose estimation. Technically, we learn a function that maps these geometric offset predictions to pairwise probabilities with logistic regression. The reason for this intermediate computation is that the multicut formulation requires calibrated pairwise costs. The pairwise affinity fields used in the algorithm of Cao *et al.* (2017) do not require such additional training step due to the use of bipartite graph matching for grouping, which is a significant advantage. PiffPaff composite fields (Kreiss *et al.*, 2019) extend vector fields of Cao *et al.* (2017) with an uncertainty estimate. Kocabas *et al.* (2018) do away without explicit pairwise terms by learning to associate keypoints to bounding box detections.

2.3 ARTICULATED POSE TRACKING

Human pose estimation in videos has been a long studied problem. Earlier work includes model-based approaches (Bregler and Malik, 1998; Sidenbladh *et al.*, 2000) which rely on tracker initialization in the first frames and are prone to drift. An alternative approach is to ignore temporal dynamics and instead find people independently in each frame so that tracking reduces to associating the detections. Such tracking-by-detection approaches (Ramanan *et al.*, 2005; Sivic *et al.*, 2005; Park and Ramanan, 2011; Fragkiadaki *et al.*, 2013) can mitigate the problem of drift. More recently, the approaches to articulated pose tracking in monocular videos relied on hand-crafted image representations and focus on simplified tasks, such as tracking upper body poses of frontal isolated people (Sapp *et al.*, 2011; Weiss and Taskar, 2013; Tokola *et al.*, 2013), or tracking walking pedestrians with little degree of articulation (Andriluka *et al.*, 2008, 2010).

The temporal dimension of videos provides a rich source of information and constraints that could be utilized to improve pose estimation, for example via temporal smoothing (Ramakrishna *et al.*, 2013; Cherian *et al.*, 2014; Zhang and Shah, 2015). Other works explore applications of optical flow for video-based pose estimation (Zuffi *et al.*, 2013; Jain *et al.*, 2014b; Pfister *et al.*, 2015). The recent work of Pfister *et al.* (2015) is based on feed-forward deep architecture that aligns predicted heatmaps of joint positions from adjacent frames and fuses them together using a learnable module. Charles *et al.* (2016) propagate annotated body keypoints using dense optical flow in order to generate annotations for a personalized model. Gkioxari *et al.* (2016) learn temporal dynamics with an auto-regressive recurrent neural network, but it considers isolated persons only and do not generalize to the case of multiple overlapping people. Other similar works (Charles *et al.*, 2016; Pfister *et al.*, 2015) consider a simplified task of tracking upper body poses of isolated upright individuals. Most of the approaches discussed so far are not directly applicable to videos with multiple potentially overlapping people.

In Chapter 4 we address a harder problem of multi-person articulated pose tracking and do not make assumptions about the type of body motions or activities of people. We take inspiration from the more complex recent models that jointly reason about entire scenes (Pishchulin *et al.*, 2016; Iqbal and Gall, 2016) as well as the approach presented in Chapter 3. However, the models are too complex and inefficient to directly generalize to image sequences. We build on the CNN detectors introduced in the Chapter 3 that are effective in localizing body joints in cluttered scenes and explore different mechanisms for assembling the joints into multiple person configurations. To that end we rely on a graph partitioning approach closely related to the prior work on human pose estimation (Pishchulin *et al.*, 2016) and multi-target object tracking (Tang *et al.*, 2015). In contrast to Tang *et al.* (2015) who focus on pedestrian tracking, and Pishchulin *et al.* (2016) who perform single frame multi-person pose estimation, we solve a more complex problem of articulated multi-person pose tracking.

Our approach is closely related to Iqbal *et al.* (2017b) who propose a similar formulation based on graph partitioning. Our approach differs from Iqbal *et al.* (2017b) primarily in the type of body-part proposals and the structure of the spatio-temporal graph. In our approach we introduce a person-conditioned model that is trained to associate body parts of a specific person already at the detection stage. This is in contrast to the approach of Iqbal *et al.* (2017b) that relies on the generic body-part detectors (Insafutdinov *et al.*, 2016a).

Our work on multi-person pose tracking spurred further research on this challenging computer vision problem. Girdhar *et al.* (2018) extends Mask R-CNN architecture of He *et al.* (2017) to videos by inflating 2D convolutional kernels to 3D (Carreira and Zisserman, 2017), such that the network produces detection "tubelets", spanning multiple frames and used Hungarian algorithm (Kuhn, 1955) to greedily associate person detections across time. Xiao *et al.* (2018) apply greedy matching as in Girdhar *et al.* (2018) and use optical flow for pose propagation as well as computing pose similarity between frames. Bertasius *et al.* (2019) train a deformable CNN (Dai *et al.*, 2017) to warp pose heatmaps between neighboring frames. At test time this enables 1) propagating sparse pose annotations between frames 2) boosting confidence of pose estimation by utilizing predictions from adjacent frames. Their approach demonstrates superior performance compared to optical flow-based confidence propagation. Raaj *et al.* (2019) propose a bottom-up method for multi-person pose tracking that extends the work of Cao *et al.* (2017). Inspired by the Part Affinity Fields (PAFs) the authors propose Temporal Affinity Fields that link keypoint detections across frames and learn them with a cascade of recurrent networks. Their method demonstrates competitive pose tracking accuracy even approaching the best top-down methods, while capable of doing online tracking in real time. Jin *et al.* (2019) extend the Associative Embedding approach of Newell *et al.* (2017) to Spatio-Temporal Embedding used for bottom-up grouping of poses in video sequences. The most recent approach of Wang *et al.* (2020) partition video sequences into overlapping clips, detects persons in a keyframe of a clip and cuts out spatio-temporal tubelets around the detected bounding boxes. The network is trained to directly predict pose tracklets of a central person in a given tubelet. Overlapping tracklets are then merged into person tracks using Hungarian algorithm. Snower *et al.* (2020) apply Transformers (Vaswani *et al.*, 2017) for performing temporal association of poses for tracking.

2.4 IMAGE- AND VIDEO-BASED DATASETS FOR HUMAN POSE ESTIMATION

Dataset	# Poses	Multi-person	Video-labeled poses	Data type
LSP Johnson and Everingham (2010)	2,000			sports (8 act.)
LSP Extended Johnson and Everingham (2011)	10,000			sports (11 act.)
MPII Single Person Andriluka *et al.* (2014)	26,429			diverse (491 act.)
FLIC Sapp and Taskar (2013)	5,003			feature movies
FashionPose Dantone *et al.* (2013)	7,305			fashion blogs
AI Challenger Wu *et al.* (2017)	700,000			diverse
We are family Eichner and Ferrari (2010)	3,131	✓		group photos
MPII Multi-Person Andriluka *et al.* (2014)	14,993	✓		diverse (491 act.)
MS COCO Keypoints Lin *et al.* (2014)	105,698	✓		diverse
OCHuman Zhang *et al.* (2019)	8,110	✓		diverse
Penn Action Zhang *et al.* (2013)	159,633		✓	sports (15 act.)
JHMDB Jhuang *et al.* (2013)	31,838		✓	diverse (21 act.)
YouTube Pose Charles *et al.* (2016)	5,000		✓	diverse
Video Pose 2.0 Sapp *et al.* (2011)	1,286		✓	TV series
Multi-Person PoseTrack Iqbal *et al.* (2017b)	16,219	✓	✓	diverse
Proposed	**276,000**	✓	✓	**diverse**

Table 2.1: Overview of publicly available datasets for articulated human pose estimation in single frames and video. For each dataset we report the number of annotated poses, availability of video pose labels and multiple annotated persons per frame, as well as types of data.

Datasets have been instrumental for the tremendous progress in the field of Computer Vision. Notably, the success of the AlexNet CNN (Krizhevsky *et al.*, 2012) would not have been possible without the large-scale ImageNet dataset (Deng *et al.*, 2009) it was trained on. Similarly, human pose estimation datasets of ever increasing size helped fuel data-driven deep learning-based algorithms. Our work on articulated pose tracking revealed the lack of a suitable video pose estimation dataset consisting of unconstrained scenes with multiple persons. In this section we review the existing datasets for human pose estimation as well as compare them to our PoseTrack dataset introduced in Chapter 5. The commonly used publicly available datasets for evaluation of 2D human pose estimation are summarized in Table 2.1. The table is split into blocks of single-person single-frame, single-person video, multi-person single-frame, and multi-person video data.

The most popular benchmarks to date for evaluation of single person pose estimation are "LSP" (Johnson and Everingham, 2010) together with "LSP Extended" (Johnson and Everingham, 2011), “MPII Human Pose (Single Person)” (Andriluka *et al.*, 2014) and MS COCO Keypoints Challenge (Lin *et al.*, 2014). LSP and LSP Extended datasets focus on sports scenes featuring a few sport types. Although a combination of both datasets results in 11,000 training poses, the evaluation set of 1000 is rather small. FLIC Sapp and Taskar (2013) targets a simpler task of upper body pose estimation of frontal upright individuals in feature movies. In contrast to LSP and FLIC datasets, MPII Single-Person benchmark covers a much wider variety of everyday human activities including various recreational, occupational and household activities and consists of over 26,000 annotated poses with 7000 poses held out for evaluation. Both benchmarks focus on single person pose estimation and provide rough location scale of a person in question. More recently, Wu *et al.* (2017) introduced “AI Challenger”: an even bigger single-frame pose estimation dataset. In contrast, our dataset addresses a much more challenging task of body tracking of multiple highly articulated individuals where neither the number of people, nor their locations or scales are known.

The single-frame multi-person pose estimation setting was introduced by Eichner and Ferrari (2010) along with “We Are Family (WAF)” dataset. While this benchmark is an important step towards more challenging multi-person scenarios, it focuses on a simplified setting of upper body pose estimation of multiple upright individuals in group photo collections. The “MPII Human Pose (Multi-Person)” dataset (Andriluka *et al.*, 2014) has significantly advanced the multi-person pose estimation task in terms of diversity and difficulty of multi-person scenes that show highly-articulated people involved in hundreds of every day activities. More recently, “MS COCO Keypoints Challenge” (Lin *et al.*, 2014) has been introduced to provide a new large-scale benchmark for single frame based multi-person pose estimation. Zhang *et al.* (2019) introduce a OCHuman – a dataset specifically targeted for heavily occluded humans with 100% instances having overlap of at least 0.5 and 32% instances with at least 0.75 overlap. The dataset contains only 8110 annotated persons and serves exclusively as val+test set to stress-test the instance segmentation and pose estimation in challenging occluded scenarios.

All these datasets are limited to single-frame body pose estimation. In contrast, our dataset also focuses on the more challenging task of multi-person pose estimation in video sequences containing highly articulated people in dense crowds. This not only requires annotations of body keypoints, but also a unique identity for every person appearing in the video. Our dataset is based on the MPII Multi-Person benchmark, from which we select a subset of key frames and for each key frame includes about five seconds of video footage centered around the key frame. We provide dense annotations of video sequences with person tracking and body pose annotations. Furthermore, we adapt a completely unconstrained evaluation setup where the scale and location of the persons is completely unknown. This is in contrast to MPII dataset that is restricted to evaluation on group crops and provides rough group location and scale. Additionally, we provide ignore regions to identify the regions containing very large crowds of people that are unreasonably complex to annotate.

Two recent works (Iqbal *et al.*, 2017b; Insafutdinov *et al.*, 2017) also provide datasets for multi-person pose estimation in videos. However, both are at a very small scale. Iqbal *et al.* (2017b) provide only 60 videos with most sequences containing only 41 frames, and Insafutdinov *et al.* (2017) provide 30 videos containing only 20 frames each. While these datasets make a first step toward solving the problem at hand, they are certainly not enough to cover a large range of real-world scenarios and to learn stronger pose estimation models. We, on the other hand, establish a large-scale benchmark with a much broader variety and an open evaluation setup. The proposed dataset contains over 270,000 annotated poses and over 46,000 labeled frames.

Our dataset is complementary to recent video datasets, such as "J-HMDB" (Jhuang *et al.*, 2013), "Penn Action" (Zhang *et al.*, 2013) and "YouTube Pose" (Charles *et al.*, 2016). Similar to these datasets, we provide dense annotations of video sequences. However, in contrast to the datasets that focus on single isolated individuals (Jhuang *et al.*, 2013; Zhang *et al.*, 2013; Charles *et al.*, 2016) we target a much more challenging task of multiple people in dynamic crowded scenarios. In contrast to YouTube Pose that focus on frontal upright people, our dataset includes a wide variety of body poses and motions, and captures people at different scales from a wide range of viewpoints. In contrast to sports-focused Penn Action and J-HMDB that focuses on a few simple actions, the proposed dataset captures a wide variety of everyday human activities while being at least 3x larger compared to J-HMDB.

Our dataset also addresses a different set of challenges compared to the datasets such as "HumanEva" by Sigal *et al.* (2010) and "Human3.6M" by Ionescu *et al.* (2013) that include images and 3D poses of people but are captured in controlled indoor environments, whereas our dataset includes real-world video sequences but provides 2D poses only.

MULTI-PERSON POSE ESTIMATION 3

Contents

In this chapter we present an approach that advances the state-of-the-art of articulated pose estimation in scenes with multiple people. To that end we contribute on three fronts. We propose (1) improved body part detectors that generate effective bottom-up proposals for body parts; (2) novel image-conditioned pairwise terms that allow to assemble the proposals into a variable number of consistent body part configurations; and (3) an incremental optimization strategy that explores the search space more efficiently thus leading both to better performance and significant speed-up factors. We evaluate our approach on two single-person and two multi-person pose estimation benchmarks. The proposed approach significantly outperforms best known multi-person pose estimation results while demonstrating competitive performance on the task of single person pose estimation

3.1 INTRODUCTION

Human pose estimation has made dramatic progress in particular on standard benchmarks for single person pose estimation (Johnson and Everingham, 2010; Andriluka *et al.*, 2014). This progress has been facilitated by the use of deep learning-based architectures (Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2014) and by the availability of large-scale datasets such as "MPII Human Pose" (Andriluka *et al.*, 2014). In order to make further progress on the challenging task of multi-person pose estimation we carefully design and evaluate several key-ingredients for human

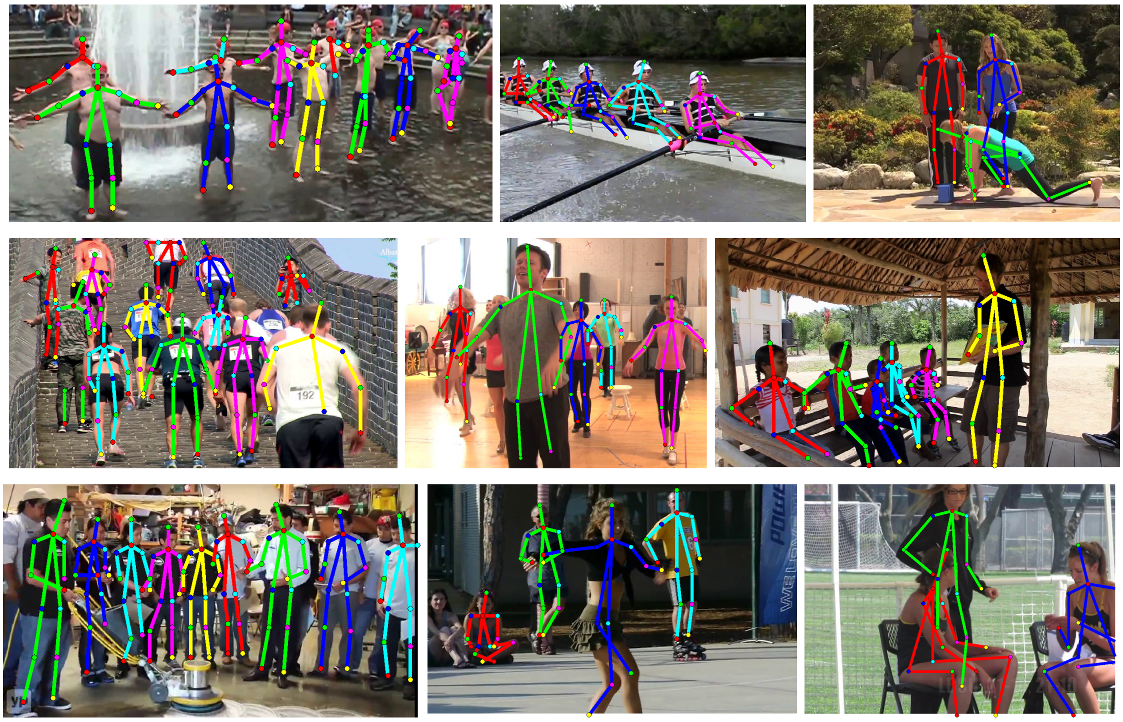


Figure 3.1: Sample multi-person pose estimation results by the proposed *DeeperCut*.

pose estimation.

The first ingredient we consider is the generation of body part hypotheses. Essentially all prominent pose estimation methods include a component that detects body parts or estimates their position. While early work used classifiers such as SVMs and AdaBoost (Johnson and Everingham, 2010; Andriluka *et al.*, 2011; Yang and Ramanan, 2012; Pishchulin *et al.*, 2013a), modern approaches build on different flavors of deep learning-based architectures (Tompson *et al.*, 2014; Chen and Yuille, 2014; Pishchulin *et al.*, 2016; Wei *et al.*, 2016). The second key ingredient are pairwise terms between body part hypotheses that help grouping those into valid human pose configurations. In earlier models such pairwise terms were essential for good performance (Johnson and Everingham, 2010; Andriluka *et al.*, 2011; Yang and Ramanan, 2012). Recent methods seem to profit less from such pairwise terms due to stronger unaries (Tompson *et al.*, 2014; Pishchulin *et al.*, 2016; Wei *et al.*, 2016). Image-conditioned pairwise terms (Pishchulin *et al.*, 2013a; Chen and Yuille, 2014) however have the promise to allow for better grouping. Last but not least, inference time is always a key consideration for pose estimation models. Often, model complexity has to be treated for speed and thus many models do not consider all spatial relations that would be beneficial for best performance.

In this chapter we discuss an approach that contributes to all three aspects and thereby significantly push the state of the art in multi-person pose estimation. We use a general optimization framework introduced by Pishchulin *et al.* (2016) as a test bed for all three key ingredients that we propose, as it allows to easily replace and combine different components. Our contributions are three-fold, leading to a novel multi-person pose estimation approach that is deeper, stronger, and faster compared to the state of the art (Pishchulin *et al.*, 2016):

- "deeper": we propose strong body part detectors based on recent advances in deep learning (He *et al.*, 2016) that – taken alone – already allow to obtain competitive performance on pose estimation benchmarks.
- "stronger": we introduce novel image-conditioned pairwise terms between body parts that allow to push performance in the challenging case of multi-people pose estimation.
- "faster": we demonstrate that using our image-conditioned pairwise along with very good part detection candidates in a fully-connected model dramatically reduces the run-time by 2–3 orders of magnitude. Finally, we introduce a novel incremental optimization method to achieve a further 4x run-time reduction while improving human pose estimation accuracy.

We evaluate our approach on two single-person and two multi-person pose estimation benchmarks and report the best results in each case. Sample multi-person pose estimation predictions by the proposed approach are shown in Figure 3.1.

3.2 MODEL

3.2.1 DeepCut Recap

Here we summarise *DeepCut* (Pishchulin *et al.*, 2016) and how unary and pairwise terms are used in this approach. *DeepCut* is a state-of-the-art approach to multi-person pose estimation based on integer linear programming (ILP) that jointly estimates poses of all people present in an image by minimizing a joint objective. This objective aims to jointly partition and label an initial pool of body part candidates into consistent sets of body-part configurations corresponding to distinct people. We use *DeepCut* as a general optimization framework that allows to easily replace and combine different components.

Specifically, *DeepCut* starts from a set D of *body part candidates*, i.e. putative detections of body parts in a given image, and a set C of *body part classes*, e.g., head, shoulder, knee. The set D of part candidates is typically generated by body part detectors and each candidate $d \in D$ has a *unary score* for every body part class $c \in C$. Based on these unary scores *DeepCut* associates a cost or reward $\alpha_{dc} \in \mathbb{R}$ to be paid by all feasible solutions of the pose estimation problem for which the body part candidate d is a body part of class c.

Additionally, for every pair of distinct body part candidates $d, d' \in D$ and every two body part classes $c, c' \in C$, the *pairwise term* is used to generate a cost or reward $\beta_{dd'cc'} \in \mathbb{R}$ to be paid by all feasible solutions of the pose estimation problem for which the body part d, classified as c, and the body part d', classified as c', belong to the same person.

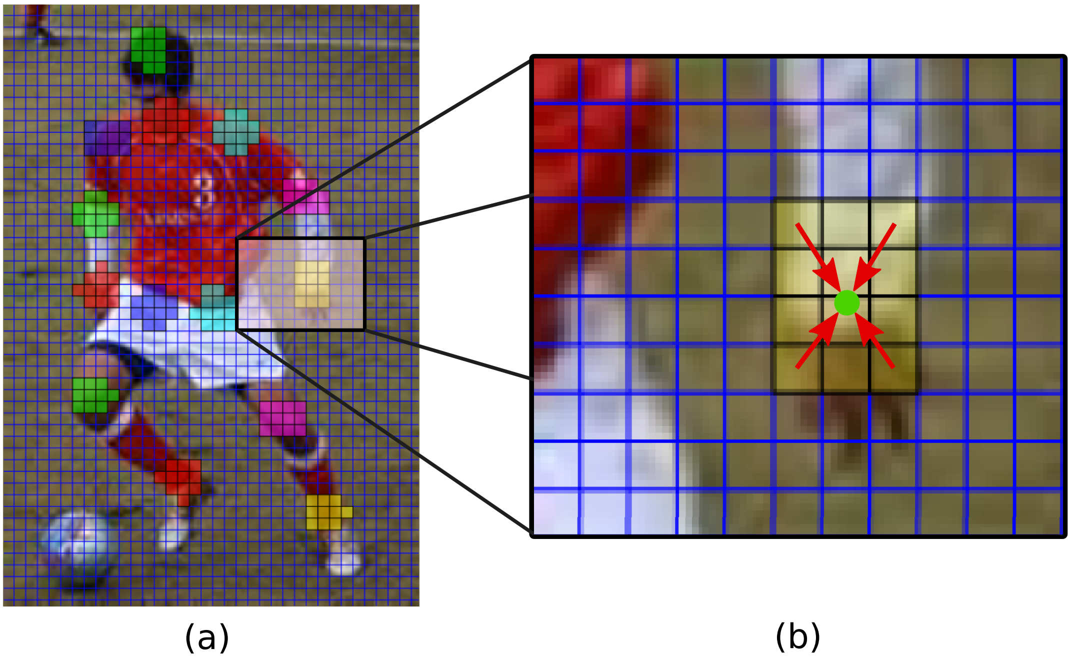


Figure 3.2: Visualisation of training ground truth for the keypoint detection network. (a) superimposed ground truth heatmaps $\hat{H}_k$ for 14 body joints. (b) visualisation of location refinement regression: we train a regressor to predict offsets $(\Delta x_k(i), \Delta y_k(j))$ (red arrows) from cells on the heatmap to the joint location (marked in green).

With respect to these sets and costs, the pose estimation problem is cast as an ILP in two classes of 01-variables: Variables $x : D \times C \to \{0,1\}$ indicate by $x_{dc} = 1$ that body part candidate d is of body part class c. If, for a $d \in D$ and all $c \in C$, $x_{dc} = 0$, the body part candidate d is suppressed. Variables $y : \binom{D}{2} \to \{0,1\}$ indicate by $y_{dd'} = 1$ that body part candidates d and d' belong to the same person. Additional variables and constraints described by Pishchulin *et al.* (2016) link the variables x and y to the costs and ensure that feasible solutions (x, y) well-define a selection and classification of body part candidates as body part classes as well as a clustering of body part candidates into distinct people.

The *DeepCut* ILP is hard and hard to approximate, as it generalizes the minimum cost multicut or correlation clustering problem which is APX-hard (Bansal *et al.*, 2004; Demaine *et al.*, 2006). Using the branch-and-cut algorithm (Pishchulin *et al.*, 2016) to compute constant-factor approximative feasible solutions of instances of the *DeepCut* ILP is not necessarily practical. In Section 3.2.5 we propose an incremental optimization approach that uses branch-and-cut algorithm to incrementally solve several instances of ILP, which results into 4–5x run-time reduction with increased pose estimation accuracy.

3.2.2 Part Detectors

As argued before, strong part detectors are an essential ingredient of modern pose estimation methods. We propose and evaluate a deep fully-convolutional human body part detection model drawing on powerful recent ideas from semantic segmentation, object classification (Long *et al.*, 2015; Chen *et al.*, 2015; He *et al.*, 2016) and human pose estimation (Tompson *et al.*, 2015; Pishchulin *et al.*, 2016; Wei *et al.*, 2016). We build our keypoint detection model on the ResNet-101 architecture (He *et al.*, 2016). Fully convolutional ResNet has stride of 32 px which is too coarse for precise part localization. We employ dilated convolutions (Chen *et al.*, 2015) in the last bank of the ResNet and an up-convolutional layer as an output layer to bring the stride of the CNN to 8 px. The first output layer of the CNN predicts a series of probability heatmaps H_k for each body part k. We use sigmoid activation function on the output neurons and binary cross entropy loss independently for each body part. We found this loss to perform better than softmax and converge much faster compared to MSE (Tompson *et al.*, 2014). Target scoremap $\hat{H}_k$ is constructed by assigning a positive label 1 at each location within a distance threshold d to the ground truth keypoint location $(\hat{x}_k^p, \hat{y}_k^p)$ of a person p present in the image, and negative label 0 otherwise:

$$\hat{H}_k(i,j) = \begin{cases} 1, & \text{if } \exists\, p : \|(i,j) \cdot s - (\hat{x}_k^p, \hat{y}_k^p)\|_2 \le d \\ 0, & \text{else,} \end{cases} \tag{3.1}$$

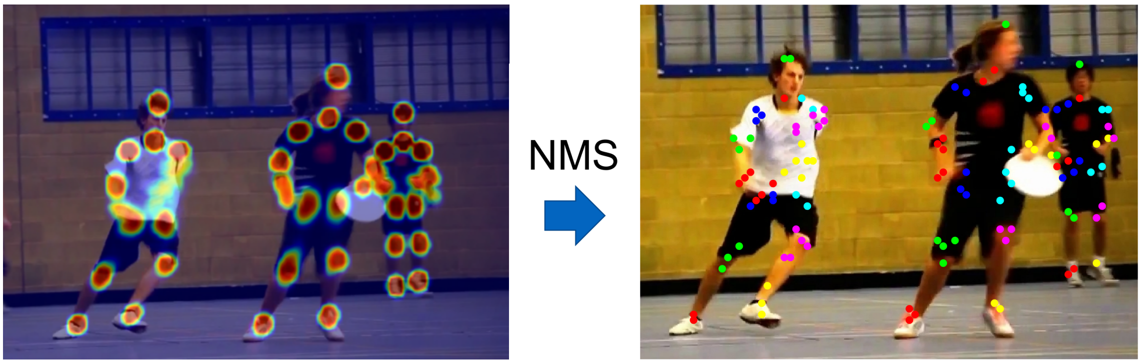
where (i,j) is the location in a downsampled heatmap and s is the stride (8px). An example of such heatmaps is demonstrated in Figure 3.2 (a).

Location refinement. In order to improve location precision we apply a technique similar to bounding box regression of Girshick (2015): we add a location refinement prediction layer and use the relative offsets $(\Delta\hat{x}_k(i), \Delta\hat{y}_k(j) = (\hat{x}_k^p - i \cdot s, \hat{y}_k^p - j \cdot s)$ from a scoremap location to the ground truth as targets, see Figure 3.2 (b). We add additional output layer to the fully convolutional CNN to predict offsets $(\Delta x_k(i), \Delta y_k(j))$ and supervise it using robust Huber loss L_H as in the work of Girshick (2015):

$$L_{loc}^k(i,j) = \begin{cases} L_{reg}^k(i,j), & \text{if } \hat{H}_k(i,j) = 1 \\ 0, & \text{else} \end{cases} \tag{3.2}$$

$$\begin{aligned} L_{reg}^k(i,j) =& L_H(\Delta x_k(i) - \Delta\hat{x}_k(i)) + \\ &+ L_H(\Delta y_k(j) - \Delta\hat{y}_k(j)) \end{aligned} \tag{3.3}$$

Eq. 3.2 ensures that the loss is only defined at locations marked as "positive" in the Eq.3.1. At test time the location (i,j) is sampled when the predicted part probability is above the certain threshold $H_k(i,j) \ge p_t$. Location refinement is then applied to compute the final keypoint coordinate: $(x_k, y_k) = (i \cdot s + \Delta x_k(i), j \cdot s + \Delta y_k(i))$.



Part heatmaps

Part proposals

Figure 3.3: Left: superimposed part probability heatmaps H_k. Right: corresponding part detections D obtained with Non-Maximum Suppression.

Receptive field size. A large receptive field size allows to incorporate context when predicting locations of individual body parts. Recent pose estimation works (Tompson *et al.*, 2014; Wei *et al.*, 2016) argue about the importance of large receptive fields and propose a complex hierarchical architecture predicting parts at multiple resolution levels. The extreme depth of ResNet allows for a very large receptive field (on the order of 1000 px compared to VGG's 400 px (Simonyan and Zisserman, 2014)) without the need of introducing complex hierarchical architectures. We empirically find that re-scaling the original image such that an upright standing person is 340 px high leads to best performance.

Intermediate supervision. Providing additional supervision addresses the problem of vanishing gradients in deep neural networks (Szegedy *et al.*, 2015; Lee *et al.*; Wei *et al.*, 2016). In addition to that, Wei *et al.* (2016) report that using part scoremaps produced at intermediate stages as inputs for subsequent stages helps to encode spatial relations between parts, while Pfister *et al.* (2015) use spatial fusion layers that learn an implicit spatial model. ResNets address the first problem by introducing identity connections and learning residual functions. To address the second concern, we make a slightly different choice: we add part loss layers inside the conv4 bank of ResNet. We argue that it is not strictly necessary to use scoremaps as inputs for the subsequent stages. The activations from such intermediate predictions are different only up to a linear transformation and contain all information about part presence that is available at that stage of the network. In Section 3.3.1 we empirically show a consistent improvement of part detection performance when including intermediate supervision.

Generating proposals. In order to incorporate the CNN output into our graph-based formulation we use Non-Maximum Suppression (NMS). This allows to convert the part probability heatmap into a finite set of body part proposals D, see Fig. 3.3. The iteration of NMS is as follows: the location (i, j) with the highest probability $H(i, j)$ is sampled, then all cells (i', j') within a certain distance to (i, j) are suppressed (unlike object detection, where boxes are suppressed based on overlap). This process

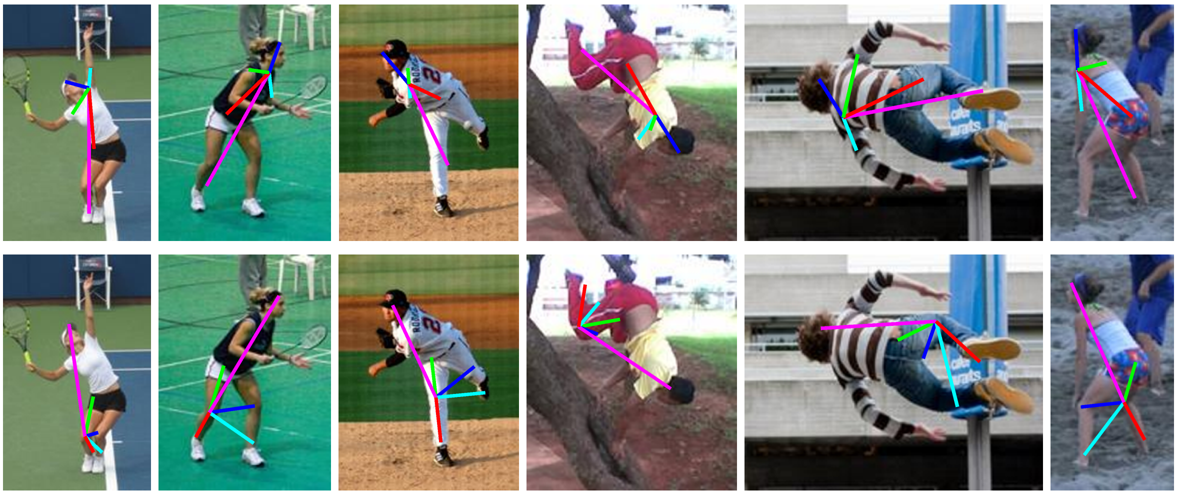


Figure 3.4: Visualizations of regression predictions. Top: from left shoulder to the right shoulder (green), right hip (red), left elbow (light blue), right ankle (purple) and top of the head (dark blue). Bottom: from right knee to the right hip (green), right ankle (red), left knee (dark blue), left ankle (light blue) and top of the head (purple). Longer-range predictions, such as e.g. shoulder – ankle may be less accurate for harder poses (top row, images 2 and 3) compared to the nearby predictions. However, they provide enough information to constrain the search space in the fully-connected spatial model.

is repeated iteratively for the remaining heatmap locations. Detections below a probability threshold $p_t = 0.1$ are not sampled.

3.2.3 Image-Conditioned Pairwise Terms

As discussed in Section 3.2.2, a large receptive field for the CNN-based part detectors allows to accurately predict the presence of a body part at a given location. However, it also contains enough evidence to reason about locations of other parts in the vicinity. We draw on this insight and propose to also use deep networks to make pairwise part-to-part predictions. They are subsequently used to compute the pairwise probabilities and show significant improvements for multi-person pose estimation.

Our approach is inspired by the body part location refinement described in Section 3.2.2. In addition to predicting offsets for the current joint, we directly regress from the current location to the relative positions of all other joints. For each scoremap location $k = (x_k, y_k)$ that is marked positive w.r.t the joint $c \in C$ and for each remaining joint $c' \in C \setminus c$, we define a relative position of c' w.r.t. c as a tuple $t^k_{cc'} = (x_{c'} - x_k, y_{c'} - x_k)$. We add an extra layer that predicts relative position $o^k_{cc'}$ and train it with a smooth L_1 loss function. We thus perform *joint* training of body part detectors (cross-entropy loss), location regression (L_1 loss) and pairwise regression (L_1 loss) by linearly combining all three loss functions. The targets t are

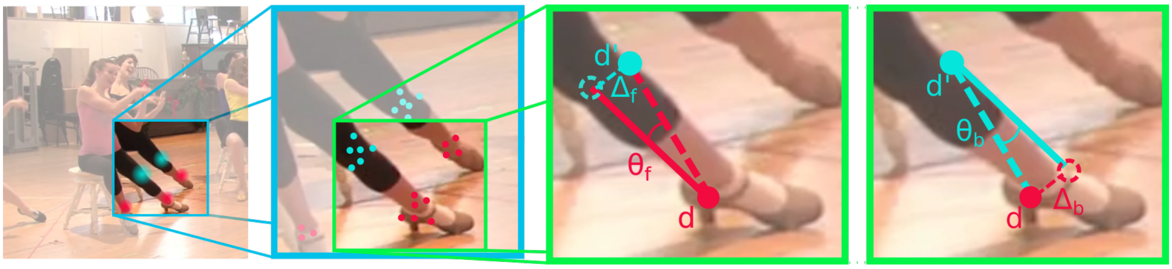


Figure 3.5: Visualization of features extracted to score the pairwise. See text for details.

normalized to have zero mean and unit variance over the training set. Results of such predictions are shown in Figure 3.4.

We then use these predictions to compute pairwise costs $\beta_{dd'cc'}$. For any pair of detections (d, d') (Figure 3.5) and for any pair of joints (c, c') we define the following quantities: locations l_d, l'_d of detections d and d' respectively; the offset prediction $o^d_{cc'}$ from c to c' at location d (solid red) coming from the CNN and similarly the offset prediction $o^{d'}_{c'c}$ (solid turquoise). We then compute the offset between the two predictions: $\hat{o}_{dd'} = l_{d'} - l_d$ (marked in dashed red). The degree to which the prediction $o^d_{cc'}$ agrees with the actual offset $\hat{o}_{dd'}$ tells how likely d, d' are of classes c, c' respectively and belong to the same person. We measure this by computing the distance between the two offsets $\Delta_f = \|\hat{o}_{dd'} - o^d_{cc'}\|_2$, and the absolute angle $\theta_f = |\measuredangle(\hat{o}_{dd'}, o^d_{cc'})|$ where f stands for forward direction, i.e from d to d'. Similarly, we incorporate the prediction $o^{d'}_{c'c}$ in the backwards direction by computing $\Delta_b = \|\hat{o}_{d'd} - o^{d'}_{c'c}\|_2$ and $\theta_b = |\measuredangle(\hat{o}_{d'd}, o^{d'}_{c'c})|$. Finally, we define a feature vector by augmenting features with exponential terms: $f_{dd'cc'} = (\Delta_f, \theta_f, \Delta_b, \theta_b, \exp(-\Delta_f), \ldots, \exp(-\theta_b))$.

We then use the features $f_{dd'cc'}$ and define logistic model:

$$p(z_{dd'cc'} = 1 | f_{dd'cc'}, \omega_{cc'}) = \frac{1}{1 + \exp(-\langle \omega_{cc'}, f_{dd'cc'} \rangle)}. \tag{3.4}$$

where $K = (|C| \times (|C| + 1))/2$ parameters $\omega_{cc'}$ are estimated using ML.

3.2.4 Sampling Detections

Location refinement NMS. *DeepCut* samples the set of detections D from the scoremap by applying non-maximum suppression (NMS). Here, we utilize location refinement and correct grid locations with the predicted offsets before applying NMS. This pulls detections that belong to a particular body joint towards its true location thereby increasing the density of detections around that location, which allows to distribute the detection candidates in a better way.

Splitting of part detections. *DeepCut* ILP solves the clustering problem by labeling each detection d with a single part class c and assigning it to a particular cluster that corresponds to a distinct person. However, it may happen that the same spatial location is occupied by more than one body joint, and therefore, its corresponding detection can only be labeled with one of the respecting classes. A naive solution is to replace a detection with n detections for each part class, which would result in a prohibitive increase in the number of detections. We simply split a detection d into several if more than one part has unary probability that is higher than a chosen threshold s (in our case $s = 0.4$).

3.2.5 Incremental Optimization

Solving one instance of the *DeepCut* ILP for all body part candidates detected for an image, as suggested by Pishchulin *et al.* (2016) and summarized in Section 3.2.1, is elegant in theory but disadvantageous in practice:

Firstly, the time it takes to compute constant-factor approximative feasible solution by the branch-and-cut algorithm (Pishchulin *et al.*, 2016) can be exponential in the number of body part candidates in the worst case. In practice, this limits the number of candidates that can be processed by this algorithm. Due to this limitation, it does happen that body parts and, for images showing many persons, entire persons are missed, simply because they are not contained in the set of candidates.

Secondly, solving one instance of the optimization problem for the entire image means that no distinction is made between part classes detected reliably, e.g. head and shoulders, and part classes detected less reliably, e.g. wrists, elbows and ankles. Therefore, it happens that unreliable detections corrupt the solution.

To address both problems, we solve not one instance of the *DeepCut* ILP but several, starting with only those body part classes that are detected most reliably and only then considering body part classes that are detected less reliably. Concretely, we study two variants of this incremental optimization approach which are defined in Table 3.5. Specifically, the procedure works as follows:

For each subset of body part classes defined in Table 3.5, an instance of the *DeepCut* ILP is set up and a constant-factor approximative feasible solution computed using the branch-and-cut algorithm. This feasible solution selects, labels and clusters a subset of part candidates, namely of those part classes that are considered in this instance. For the next instance, each cluster of body part candidates of the same class from the previous instance becomes just one part candidate whose class is fixed. Thus, the next instance is an optimization problem for selecting, labeling and clustering body parts that have not been determined by previous instances. Overall, this allows us to start with more part candidates consistently and thus improve the pose estimation result significantly.

3.3 EXPERIMENTS

Implementation details. We use the publicly available ResNet implementation (Caffe) and initialize from the ImageNet-pre-trained models. We train networks with SGD for 1M iterations, starting with the learning rate lr=0.001 for 10k, then lr=0.002

for 420k, lr=0.0002 for 300k and lr=0.0001 for 300k. This corresponds to roughly 17 epochs of the MPII (Andriluka *et al.*, 2014) train set. Finetuning from ImageNet takes two days on a *single* GPU. Batch normalization (Ioffe and Szegedy) worsens performance, as the batch size of 1 in fully convolutional training is not enough to provide a reliable estimate of activation statistics. During training we switch off collection of statistics and use the mean and variance that were gathered on the ImageNet dataset.

3.3.1 Evaluation of Part Detectors

Datasets. We use three public datasets: "Leeds Sports Poses" (LSP) (Johnson and Everingham, 2010) (person-centric (PC) annotations); "LSP Extended" (LSPET) (Johnson and Everingham, 2011); "MPII Human Pose" ("Single Person") (Andriluka *et al.*, 2014) consisting of 19185 training and 7247 testing poses. To evaluate on LSP we train part detectors on the union of MPII, LSPET and LSP training sets. To evaluate on MPII Single Person we train on MPII *only*.

Evaluation measures. We use the standard "Percentage of Correct Keypoints (PCK)" evaluation metric (Sapp and Taskar, 2013; Toshev and Szegedy, 2014; Tompson *et al.*, 2014) and evaluation scripts from the web page of MPII Pose dataset (Andriluka *et al.*, 2014). In addition to PCK at fixed threshold, we report "Area under Curve" (AUC) computed for the entire range of PCK thresholds.

Results on LSP. The results are shown in Table 3.1. ResNet-50 with 8 px stride achieves 87.8% PCK and 63.7% AUC. Increasing the stride size to 16 px and up-sampling the scoremaps by 2x to compensate for the loss on resolution slightly drops the performance to 87.2% PCK. This is expected as up-sampling cannot fully compensate for the information loss due to a larger stride. Larger stride minimizes memory requirements, which allows for training a deeper ResNet-152. The latter significantly increases the performance (89.1 vs. 87.2% PCK, 65.1 vs. 63.1% AUC), as it has larger model capacity. Introducing intermediate supervision further improves the performance to 90.1% PCK and 66.1% AUC, as it constraints the network to learn useful representations in the early stages and uses them in later stages for spatial disambiguation of parts.

The results are compared to the state of the art in Table 3.1. Our best model significantly outperforms *DeepCut* (Pishchulin *et al.*, 2016) (90.1% PCK vs. 87.1% PCK), as it relies on deeper detection architectures. Our model performs on par with the recent approach of Wei *et al.* (2016) (90.1 vs. 90.5% PCK, 66.1 vs. 65.4 AUC). This is interesting, as they use a much more complex multi-scale multi-stage architecture.

Results on MPII Single Person. The results are shown in Table 3.2. ResNet-152 achieves 87.8% PCK$_h$ and 60.0% AUC, while intermediate supervision slightly improves the performance further to 88.5% PCK$_h$ and 60.8% AUC. Comparing the results to the state of the art we observe significant improvement over *DeepCut* (Pishchulin *et al.*, 2016) (+5.9% PCK$_h$, +4.2% AUC), which again underlines the importance of using extremely deep model. The proposed approach performs

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK	AUC
ResNet-50 (8 px)	96.9	90.3	85.0	81.5	88.6	87.3	84.8	87.8	63.7
ResNet-50 (16 px + 2x up-sample)	96.7	89.8	84.6	80.4	89.3	86.4	82.8	87.2	63.1
ResNet-101 (16 px + 2x up-sample)	96.9	91.2	85.8	82.6	90.9	**90.2**	85.9	89.1	64.6
ResNet-152 (16 px + 2x up-sample)	97.4	91.7	85.7	82.4	90.1	89.2	86.9	89.1	65.1
+ intermediate supervision	97.4	**92.7**	**87.5**	**84.4**	**91.5**	89.9	87.2	90.1	**66.1**
DeepCut (Pishchulin *et al.*, 2016)	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1	63.5
Wei *et al.* (2016)	**97.8**	92.5	87.0	83.9	**91.5**	90.8	**89.9**	**90.5**	65.4
Tompson *et al.* (2014)	90.6	79.2	67.9	63.4	69.5	71.0	64.2	72.3	47.3
Chen and Yuille (2014)	91.8	78.2	71.8	65.5	73.3	70.2	63.4	73.4	40.1
Fan *et al.* (2015)	92.4	75.2	65.3	64.0	75.7	68.3	70.4	73.0	43.2

Table 3.1: Pose estimation results (PCK) on LSP (PC) dataset.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	PCK$_h$	AUC
ResNet-152	96.3	94.1	88.6	83.9	87.2	82.9	77.8	87.8	60.0
+ intermediate supervision	96.8	**95.2**	**89.3**	**84.4**	**88.4**	**83.4**	78.0	**88.5**	60.8
DeepCut (Pishchulin *et al.*, 2016)	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4	56.5
Tompson *et al.* (2014)	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6	51.8
Carreira *et al.* (2016)	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3	49.1
Tompson *et al.* (2015)	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0	54.9
Wei *et al.* (2016)	**97.8**	95.0	88.7	84.0	**88.4**	82.8	**79.4**	**88.5**	**61.4**

Table 3.2: Pose estimation results (PCK$_h$) on MPII Single Person.

on par with the best know result by Wei *et al.* (2016) (88.5 vs. 88.5% PCK$_h$) for the maximum distance threshold, while slightly loosing when using the entire range of thresholds (60.8 vs. 61.4% AUC). We envision that extending the proposed approach to incorporate multiple scales as in the work of Wei *et al.* (2016) should improve the performance. The model trained on the union of MPII, LSPET and LSP training sets achieves 88.3% PCK$_h$ and 60.7% AUC. The fact that the same model achieves similar performance on both LSP and MPII benchmarks demonstrates the generality of our approach.

3.3.2 Evaluation of Pairwise Terms

Datasets and evaluation measure. We evaluate on the challenging public "MPII Human Pose" ("Multi-Person") benchmark (Andriluka *et al.*, 2014) consisting of 3844

Unary	Pairwise	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	time [s/frame]
DeepCut	*DeepCut*	50.1	44.1	33.5	26.5	33.0	28.5	14.4	33.3	259220
DeepCut	this work	68.3	58.3	47.4	38.9	45.2	41.8	31.2	47.7	1987
this work	this work	**70.9**	59.8	**53.1**	**44.4**	50.0	46.4	39.5	52.3	1171
+ location refinement before NMS		70.3	**61.6**	52.1	43.7	**50.6**	**47.0**	**40.6**	**52.6**	**578**

Table 3.3: Effects of proposed pairwise and unaries on the pose estimation performance (AP) on MPII Multi-Person Val and comparison with the *DeepCut* model of Pishchulin *et al.* (2016).

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	time [s/frame]
bi-directional + angle	**70.3**	**61.6**	**52.1**	43.7	**50.6**	**47.0**	**40.6**	**52.6**	**578**
uni-directional + angle	69.3	58.4	51.8	**44.2**	50.4	44.7	36.3	51.1	2140
bi-directional	68.8	58.3	51.0	42.7	51.1	46.5	38.7	51.3	914

Table 3.4: Effects of different versions of the pairwise terms on the pose estimation performance (AP) on MPII Multi-Person Val.

training and 1758 testing groups of multiple overlapping people in highly articulated poses with a variable number of parts. We perform all intermediate experiments on a validation set of 200 images sampled uniformly at random and refer to it as MPII Multi-Person Val. We report major results on the full testing set, and on the subset of 288 images for the direct comparison to the work of Pishchulin *et al.* (2016). The AP measure (Pishchulin *et al.*, 2016) evaluating consistent body part detections is used for performance comparison. Additionally, we report median running time per frame measured in seconds[1].

Evaluation of unaries and pairwise. The results are shown in Table 3.3. Baseline *DeepCut* achieves 33.3% AP. Using the proposed pairwise significantly improves performance achieving 47.7% AP. This clearly shows the advantages of using image-conditioned pairwise to disambiguate the body part assignment for multiple overlapping individuals. Remarkably, the proposed pairwise dramatically reduce the run-time by two orders of magnitude (1987 vs. 259220 s/frame). This underlines the argument that using strong pairwise in the fully-connected model allows to significantly speed-up the inference. Using additionally the proposed part detectors further boosts the performance (52.3 vs. 47.7% AP), which can be attributed to better quality part hypotheses. Run-time is again almost halved, which clearly shows the importance of obtaining high-quality part detection candidates for more accurate and faster inference. Performing location refinement before NMS slightly improves the performance, but also reduces the run-time by 2x: this allows to increase the density of detections at the most probable body part locations and thus suppresses

[1] Run-time is measured on a single core Intel Xeon 2.70GHz

	Stage 1	Stage 2	Stage 3
2-stage	head, shoulders elbows, wrists	hips, knees ankles	
3-stage	head shoulders	elbows wrists	hips, knees ankles

Table 3.5: As the run-time of the DeepCut branch-and-cut algorithm limits the number of part candidates that can be processed in practice, we split the set of part classes into subsets, coarsely and finely, and solve the pose estimation problem incrementally.

more detections around the most confident ones, which leads to better distribution of part detection candidates and reduces confusion generated by the near-by detections. Overall, we observe significant performance improvement and dramatic reduction in run-time by the proposed *DeeperCut* compared to the baseline *DeepCut*.

Ablation study of pairwise. An ablation study of the proposed image-conditioned pairwise is performed in Table 3.4. Regressing from both joints onto the opposite joint's location and including angles achieves the best performance of 52.6% AP and the minimum run-time of 578 s/frame. Regressing from a single joint only slightly reduces the performance to 51.1% AP, but significantly increases run-time by 4x: these pairwise are less robust compared to the bi-directional, which confuses the inference. Removing the angles from the pairwise features also decreases the performance (51.3 vs. 52.6% AP) and doubles run-time, as it removes the information about body part orientation.

3.3.3 Evaluation of Incremental Optimization

Results are shown in Table 3.6. Single stage optimization with $|D| = 100$ part detection candidates achieves 52.6% AP (best from Table 3.3). More aggressive NMS with radius of 24 px improves the performance (54.5 vs. 52.6% AP), as it allows to better distribute detection candidates. Increasing $|D|$ to 150 slightly improves the performance by +0.6% AP, but significantly increases run-time (1041 vs. 596 s/frame). We found $|D| = 150$ to be maximum total number of detection candidates (11 per part) for which optimization runs in a reasonable time. Incremental optimization of 2-stage inference slightly improves the performance (56.5 vs. 55.1% AP) as it allows for a larger number of detection candidates per body part (20) and leverages typically more confident predictions of the upper body parts in the first stage before solving for the entire body. Most importantly, it halves the median run-time from 1041 to 483 s/frame. Incremental optimization of 3-stage inference again almost halves the run-time to 271 s/frame while noticeably improving the human pose estimation performance for all body parts but elbows achieving 57.6% AP. These results clearly

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	time [s/frame]
1-stage optimize, 100 det, nms 1x	70.3	61.6	52.1	43.7	50.6	47.0	40.6	52.6	578
1-stage optimize, 100 det, nms 2x	71.3	64.1	55.8	44.1	53.8	48.7	41.3	54.5	596
1-stage optimize, 150 det, nms 2x	74.1	65.6	56.0	44.3	54.4	49.2	39.8	55.1	1041
2-stage optimize	75.9	66.8	58.8	46.1	54.1	48.7	42.4	56.5	483
3-stage optimize	78.3	69.3	58.4	47.5	55.1	49.6	42.5	57.6	271
+ split detections	**78.5**	**70.5**	**59.7**	**48.7**	**55.4**	**50.6**	**44.4**	**58.7**	**270**
DeepCut Pishchulin *et al.* (2016)	50.1	44.1	33.5	26.5	33.0	28.5	14.4	33.3	259220

Table 3.6: Performance (AP) of different hierarchical versions of *DeeperCut* on MPII Multi-Person Val.

demonstrate the advantages of the proposed incremental optimization. Splitting the detection candidates that simultaneously belong to multiple body parts with high confidence slightly improves the performance to 58.7% AP. This helps to overcome the limitation that each detection candidate can be assigned to a single body part and improves on cases where two body parts overlap thus sharing the same detection candidate. We also compare the obtained results to *DeepCut* in Table 3.6 (last row). The proposed *DeeperCut* outperforms baseline *DeepCut* (58.7 vs. 33.3% AP) by almost doubling the performance, while run-time is reduced dramatically by 3 orders of magnitude from the infeasible 259220 s/frame to affordable 270 s/frame. This comparison clearly demonstrates the power of the proposed approach and dramatic effects of better unary, pairwise and optimization on the overall pose estimation performance and run-time.

3.3.4 Comparison to the State of the Art

We compare to other works on MPII Multi-Person Test and WAF (Eichner and Ferrari, 2010) datasets.

Results on MPII Multi-Person. For direct comparison with *DeepCut* we evaluate on the same subset of 288 testing images as in the work of Pishchulin *et al.* (2016). Additionally, we provide the results on the entire testing set. Results are shown in Table 3.7. *DeeperCut* without incremental optimization already outperforms *DeepCut* by a large margin (66.2 vs. 54.1% AP). Using 3-stage incremental optimization further improves the performance to 69.7% AP improving by a dramatic 16.5% AP over the baseline. Remarkably, the run-time is reduced from 57995 to 230 s/frame, which is an improvement by two orders of magnitude. Both results underline the importance of strong image-conditioned pairwise terms and incremental optimization to maximize multi-person pose estimation performance at the reduced run-time. A similar trend is observed on the full set: 3-stage optimization improves over a single stage optimization (59.4 vs. 54.7% AP). We observe that the performance on the entire

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	time [s/frame]
subset of 288 images as in Pishchulin *et al.* (2016)									
DeeperCut (1-stage)	83.3	79.4	66.1	57.9	63.5	60.5	49.9	66.2	1333
DeeperCut	**87.5**	**82.8**	**70.2**	**61.6**	**66.0**	**60.6**	**56.5**	**69.7**	**230**
DeepCut	73.4	71.8	57.9	39.9	56.7	44.0	32.0	54.1	57995
full set									
DeeperCut (1-stage)	73.7	65.4	54.9	45.2	52.3	47.8	40.7	54.7	2785
DeeperCut	**79.1**	**72.2**	**59.7**	**50.0**	**56.0**	**51.0**	**44.6**	**59.4**	485
Faster R-CNN + unary	64.9	62.9	53.4	44.1	50.7	43.1	35.2	51.0	**1**

Table 3.7: Pose estimation results (AP) on MPII Multi-Person.

Setting	Head	U Arms	L Arms	Torso	*m*PCP	AOP
DeeperCut nms 3.0	**99.3**	**83.8**	**81.9**	**87.1**	**86.3**	**88.1**
DeepCut (Pishchulin *et al.*, 2016)	**99.3**	81.5	79.5	**87.1**	84.7	86.5
Ghiasi *et al.* (2014)	-	-	-	-	63.6	74.0
Eichner and Ferrari (2010)	97.6	68.2	48.1	86.1	69.4	80.0
Chen and Yuille (2015)	98.5	77.2	71.3	88.5	80.7	84.9

Table 3.8: Pose estimation results (*m*PCP) on WAF dataset.

testing set is over 10% AP lower compared to the subset and run-time is doubled. This implies that the subset of 288 images is easier compared to the full testing set. We envision that performance differences between *DeeperCut* and *DeepCut* on the entire set will be at least as large as when compared on the subset. We also compare to a strong two-stage baseline: first each person is pre-localized by applying the state-of-the-art detector (Ren *et al.*, 2015) following by NMS and retaining rectangles with scores at least 0.8; then pose estimation for each rectangle is performed using *DeeperCut* unary only. Being significantly faster (1 s/frame) this approach reaches 51.0% AP vs. 59.4% AP by *DeeperCut*, which clearly shows the power of joint reasoning by the proposed approach.

Results on WAF. Results using the official evaluation protocol (Eichner and Ferrari, 2010) assuming *m*PCP and AOP evaluation measures and considering detection bounding boxes provided by Eichner and Ferrari (2010) are shown in Table 3.8. *DeeperCut* achieves the best result improving over the state of the art *DeepCut* (86.3 vs. 84.7% *m*PCP, 88.1 vs. 86.5% AOP). Noticeable improvements are observed both for upper (+2.3% *m*PCP) and lower (+2.4% *m*PCP) arms. However, overall performance differences between *DeeperCut* and the baseline *DeepCut* are not as pronounced compared to MPII Multi-Person dataset. This is due to the fact that actual differences

Setting	Head	Sho	Elb	Wri	AP	time [s/frame]
DeeperCut	**92.6**	**81.1**	**75.7**	**78.8**	**82.0**	**13**
DeepCut (Pishchulin *et al.*, 2016)	76.6	80.8	73.7	73.6	76.2	22000
Chen and Yuille (2015)	83.3	56.1	46.3	35.5	55.3	-

Table 3.9: Pose estimation results (AP) on WAF dataset.

are washed out by the peculiarities of the *m*PCP evaluation measure: *m*PCP assumes that people are pre-detected and human pose estimation performance is evaluated only for people whose upper body detections match the ground truth. Thus, a pose estimation method is not penalized for generating multiple body pose predictions, since the only pose prediction is considered whose upper body bounding box best matches the ground truth. We thus re-evaluate the competing approaches (Pishchulin *et al.*, 2016; Chen and Yuille, 2015) using the more realistic AP evaluation measure[2]. The results are shown in Table 3.9. *DeeperCut* significantly improves over *DeepCut* (82.0 vs. 76.2% AP). The largest boost in performance is achieved for head (+16.0% AP) and wrists (+5.2% AP): *DeeperCut* follows incremental optimization strategy by first solving for the most reliable body parts, such as head and shoulders, and then using the obtained solution to improve estimation of harder body parts, such as wrists. Most notably, run-time is dramatically reduced by 3 orders of magnitude from 22000 to 13 s/frame. These results clearly show the advantages of the proposed approach when evaluated in the real-world detection setting. The proposed *DeeperCut* also outperforms the work of Chen and Yuille (2015) by a large margin. The performance difference is much more pronounced compared to using *m*PCP evaluation measure: in contrast to *m*PCP, AP penalizes multiple body pose predictions of the same person. We envision that better NMS strategies are likely to improve the AP performance of Chen and Yuille (2015).

3.4 CONCLUSION

In this chapter we presented an articulated multi-person 2D pose estimation system that significantly advanced the state of the art. To that end we carefully re-designed and thoroughly evaluated several key ingredients. First, drawing on the recent advances in deep learning we proposed strong extremely deep body part detectors that – taken alone – already allow to obtain state of the art performance on standard pose estimation benchmarks. Second, we introduce novel image-conditioned pairwise terms between body parts that allow to significantly push the performance in the challenging case of multi-people pose estimation, and dramatically reduce the run-time of the inference in the fully-connected spatial model. Third, we introduced a

[2] We used publicly-available pose predictions of Chen and Yuille (2015) for all people in WAF dataset.

novel incremental optimization strategy to further reduce the run-time and improve human pose estimation accuracy. Overall, the proposed improvements allowed to almost double the pose estimation accuracy in the challenging multi-person case while reducing the run-time by 3 orders of magnitude.

ARTICULATED MULTI-PERSON TRACKING 4

Contents

In this chapter we propose an approach for articulated tracking of multiple people in unconstrained videos. Our starting point is a model that resembles existing architectures for single-frame pose estimation but is substantially faster. We achieve this in two ways: (1) by simplifying and sparsifying the body-part relationship graph and leveraging recent methods for faster inference, and (2) by offloading a substantial share of computation onto a feed-forward convolutional architecture that is able to detect and associate body joints of the same person even in clutter. We use this model to generate proposals for body joint locations and formulate articulated tracking as spatio-temporal grouping of such proposals. This allows to jointly solve the association problem for all people in the scene by propagating evidence from strong detections through time and enforcing constraints that each proposal can be assigned to one person only. We report results on a public “MPII Human Pose” benchmark and on a new “MPII Video Pose” dataset of image sequences with multiple people. We demonstrate that our model achieves state-of-the-art results while using only a fraction of time and is able to leverage temporal information to improve state-of-the-art for crowded scenes.

4.1 INTRODUCTION

Here, we address the task of articulated human pose tracking in monocular video. We focus on scenes of realistic complexity that often include fast motions, large

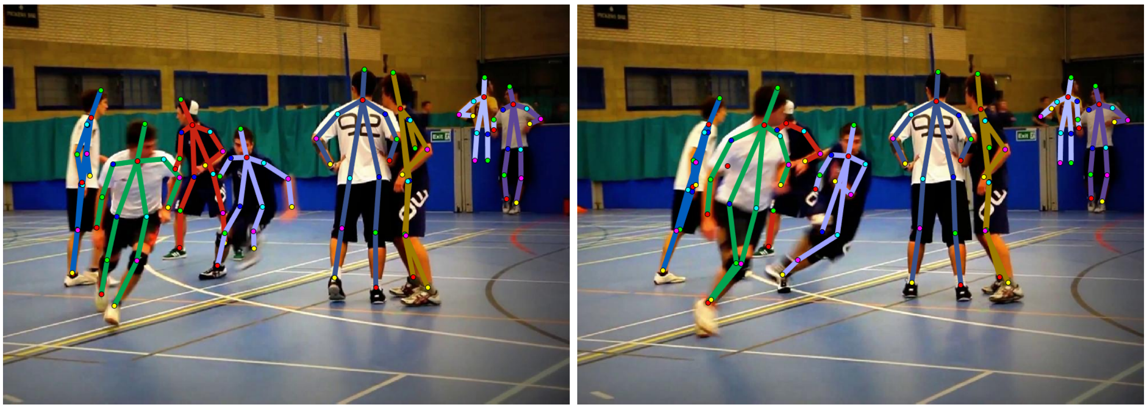


Figure 4.1: Example articulated tracking results of our approach.

variability in appearance and clothing, and person-person occlusions. A successful approach must thus identify the number of people in each video frame, determine locations of the joints of each person and associate the joints over time.

One of the key challenges in such scenes is that people might overlap and only a subset of joints of the person might be visible in each frame either due to person-person occlusion or truncation by image boundaries (Fig. 4.1). Arguably, resolving such cases correctly requires reasoning beyond purely geometric information on the arrangement of body joints in the image, and requires incorporation of a variety of image cues and joint modeling of several persons.

The design of our model is motivated by two factors. We would like to leverage bottom-up end-to-end learning to directly capture image information. At the same time we aim to address a complex multi-person articulated tracking problem that does not naturally lend itself to an end-to-end prediction task and for which training data is not available in the amounts usually required for end-to-end learning.

To leverage the available image information we learn a model for associating a body joint to a specific person in an end-to-end fashion relying on a convolutional network. We then incorporate these part-to-person association responses into a framework for jointly reasoning about assignment of body joints within the image and over time. To that end we use the graph partitioning formulation that has been used for people tracking and pose estimation in the past (Tang *et al.*, 2015; Pishchulin *et al.*, 2016), but has not been shown to enable articulated people tracking.

To facilitate efficient inference in video we resort to fast inference methods based on local combinatorial optimization (Levinkov *et al.*, 2017) and aim for a sparse model that keeps the number of connections between variables to a minimum. As we demonstrate, in combination with feed-forward reasoning for joint-to-person association this allows us to achieve substantial speed-ups compared to state-of-the-art (Insafutdinov *et al.*, 2016a) while maintaining the same level of accuracy.

Our main contribution is a new articulated tracking model that operates by bottom-up assembly of part detections within each frame and over time. In contrast to recent works (Gkioxari *et al.*, 2016; Pfister *et al.*, 2015) this model is suitable for scenes with an unknown number of subjects and reasons jointly across multiple people incorporating inter-person exclusion constraints and propagating strong observations to neighboring frames.

Our second contribution is a formulation for single-frame pose estimation that relies on a sparse graph between body parts and a mechanism for generating body-part proposals conditioned on a person's location. This is in contrast to state-of-the-art approaches (Pishchulin *et al.*, 2016; Insafutdinov *et al.*, 2016a) that perform expensive inference in a full graph and rely on generic bottom-up proposals. We demonstrate that a sparse model with a few spatial edges performs competitively with a fully-connected model while being much more efficient. Notably, a simple model that operates in top-down/bottom-up fashion exceeds the performance of a fully-connected model while being 24x faster at inference time (cf. Tab. 4.3). This is due to offloading of a large share of the reasoning about body-part association onto a feed-forward convolutional architecture.

4.1.1 Overview

Our model consists of the two components: (1) a convolutional network for generating body part proposals and (2) an approach to group the proposals into spatio-temporal clusters. In Sec. 4.2 we introduce a general formulation for multi-target tracking that follows the work by Tang *et al.* (2015) and allows us to define pose estimation and articulated tracking in a unified framework. We then describe the details of our articulated tracking approach in Sec. 4.3, and introduce two variants of our formulation: bottom-up (*BU*) and top-down/bottom-up (*TD/BU*). We present experimental results in Sec. 4.4.

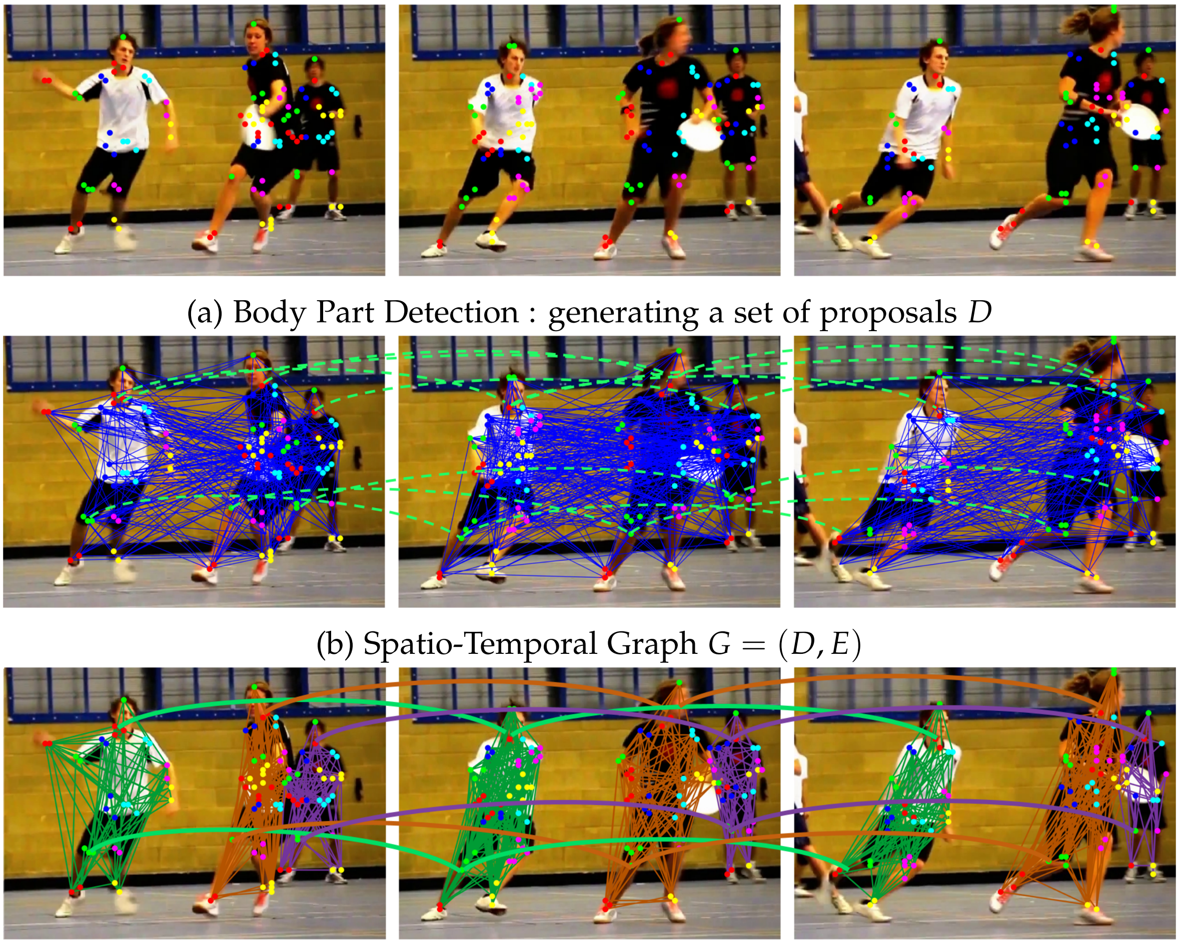
4.2 TRACKING BY SPATIO-TEMPORAL GROUPING

Our body part detector generates a set of proposals $D = \{\mathbf{d}_i\}$ for each frame of the video (Figure 4.2 (a)). Each proposal is given by $\mathbf{d}_i = (t_i, d_i^{pos}, \pi_i, \tau_i)$, where t_i denotes the index of the video frame, d_i^{pos} is the spatial location of the proposal in image coordinates, π_i is the probability of correct detection, and τ_i is the type of the body joint (*e.g.* ankle or shoulder).

Let $G = (D, E)$ be a graph whose nodes D are the joint detections in a video and whose edges E connect pairs of detections that hypothetically correspond to the same target (Figure 4.2 (b)).

The output of the tracking algorithm is a subgraph $G' = (D', E')$ of G, where D' is a subset of nodes after filtering redundant and erroneous detections and E' are edges linking nodes corresponding to the same target. We specify G' via binary variables $x \in \{0,1\}^D$ and $y \in \{0,1\}^E$ that define subsets of edges and nodes included in G'. In particular each track will correspond to a connected component in G'.

As a general way to introduce constraints on edge configurations that correspond to a valid tracking solution we introduce a set $Z \subseteq \{0,1\}^{D \cup E}$ and define a combination of edge and node indicator variables to be feasible if and only if $(x, y) \in Z$.



(a) Body Part Detection : generating a set of proposals D

(b) Spatio-Temporal Graph $G = (D, E)$

(c) Spatio-Temporal grouping: find $G' = (D', E') \subset G$

Figure 4.2: Tracking by spatio-temporal grouping. (a) shows the initial set D of body part proposals. (b) shows spatio temporal graph G. Edges that connect proposals within frame are displayed in blue. They include *cross-type* edges that connect proposals of different types (for instance head-shoulder), and *same-type* (attractive-repulsive) edges, that connect nodes of the same type. *Temporal* edges connecting nodes in neighboring frames are displayed dashed green. (c) shows the result of our tracking approach: after partitioning of G each connected component (colored in green, brown and purple) corresponds to an individual person.

An example of a constraint encoded through Z is that endpoint nodes of an edge included by y must also be included by x. Note that the variables x and y are coupled though Z. Moreover, assuming that $(x, y) \in Z$ we are free to set components of x and y independently to maximize the tracking objective.

Given image observations we compute a set of features for each node and edge in the graph. We denote such node and edge features as f and g respectively. Assuming independence of the feature vectors the conditional probability of indicator functions x of nodes and y of edges given features f and g and given a feasible set Z is given by

$$p(x, y|f, g, Z) \propto p(Z|x, y) \prod_{d \in D} p(x_d|f^d) \prod_{e \in E} p(y_e|g^e), \tag{4.1}$$

where $p(Z|x, y)$ assigns a constant non-zero probability to every feasible solution and is equal to zero otherwise. Minimizing the negative log-likelihood of Eq. 4.1 is equivalent to solving the following integer-linear program:

$$\min_{(x,y) \in Z} \sum_{d \in D} c_d x_d + \sum_{e \in E} d_e y_e \ , \tag{4.2}$$

where $c_d = \log \frac{p(x_d=1|f^d)}{p(x_d=0|f^d)}$ is the cost of retaining d as part of the solution, and $d_e = \log \frac{p(y_e=1|g^e)}{p(y_e=0|g^e)}$ is the cost of assigning the detections linked by an edge e to the same track.

We define the set of constraints Z as in Tang *et al.* (2015):

$$\forall e = vw \in E : \quad y_{vw} \leq x_v \tag{4.3}$$

$$\forall e = vw \in E : \quad y_{vw} \leq x_w \tag{4.4}$$

$$\forall C \in \text{cycles}(G) \ \forall e \in C : \quad (1 - y_e) \leq \sum_{e' \in C \setminus \{e\}} (1 - y_{e'}) \tag{4.5}$$

Jointly with the objective in Eq. 4.2 the constraints (4.3)-(4.5) define an instance of the minimum cost subgraph multicut problem (Tang *et al.*, 2015). The constraints (4.3) and (4.4) ensure that assignment of node and edge variables is consistent. The constraint (4.5) ensures that for every two nodes either all or none of the paths between these nodes in graph G are contained in one of the connected components of subgraph G'. This constraint is necessary to unambigously assign person identity to a body part proposal based on its membership in a specific connnected component of G'.

4.3 ARTICULATED MULTI-PERSON TRACKING

In Section 4.2 we introduced a general framework for multi-object tracking by solving an instance of the subgraph multicut problem. The subgraph multicut problem is

NP-hard, but recent work (Tang *et al.*, 2015; Levinkov *et al.*, 2017) has shown that efficient approximate inference is possible with local search methods. The framework allows for a variety of graphs and connectivity patterns. Simpler connectivity allows for faster and more efficient processing at the cost of ignoring some of the potentially informative dependencies between model variables. Our goal is to design a model that is efficient, with as few edges as possible, yet effective in crowded scenes, and that allows us to model temporal continuity and inter-person exclusion. Our articulated tracking approach proceeds by constructing a graph G that couples body part proposals within the same frame and across neighboring frames. In general the graph G will have three types of edges, shown in Figure 4.2 (b): (1) *cross-type* edges that connect two parts of different types, (2) *same-type* edges that connect two nodes of the same type in the same image, and (3) *temporal* edges that connect nodes in the neighboring frames.

We now define two variants of our model that we denote as *Bottom-Up* (*BU*) and *Top-Down/Bottom-Up* (*TD/BU*). In the *BU* model the body part proposals are generated with the convolutional part detector described in Section 3.2.2. In the *TD/BU* model we substitute these generic part detectors with a new convolutional body-part detector that is trained to output consistent body configurations conditioned on the person location. This alows to further reduce the complexity of the model graph since the task of associating body parts is addressed within the proposal mechanism. As we show in Section 4.4 this leads to considerable gains in performance and allows for faster inference. Note that the *BU* and *TD/BU* models have identical *same-type* and *temporal* pairwise terms, but differ in the form of *cross-type* pairwise terms, and the connectivity of the nodes in G. For both models we rely on the solver from the work of Levinkov *et al.* (2017) for inference.

4.3.1 Bottom-Up Model (*BU*).

For each body part proposal $\mathbf{d}_i$ the detector outputs image location, probability of detection π_i, and a label τ_i that indicates the type of the detected part (*e.g.* shoulder or ankle). We directly use the probability of detection to derive the unary costs in Eq. 4.2 as $c_{d_i} = \log(\pi_i/(1-\pi_i))$. Image features f^d in this case correspond to the image representation generated by the convolutional network.

We consider two connectivity patterns for nodes in the graph G. We either define edges for every pair of body part types, as is shown in Figure 4.3 (a), which results in a fully connected graph in each image. Alternatively we obtain a sparse version of the model by defining edges for a subset of part types only (Figure 4.3 (b)). The rationale behind the sparse version is to obtain a simpler and faster version of the model by omitting edges between parts that carry little information about each other's image location (*e.g.* left ankle and right arm).

Edge costs. In our *Bottom-Up* model the cost of the edges d_e connecting two body part detections $\mathbf{d}_i$ and $\mathbf{d}_j$ is defined as a function of the detection types τ_i and τ_j. Following Insafutdinov *et al.* (2016a) we thus train for each pair of part types a regression function that predicts relative image location of the parts in the pair. The

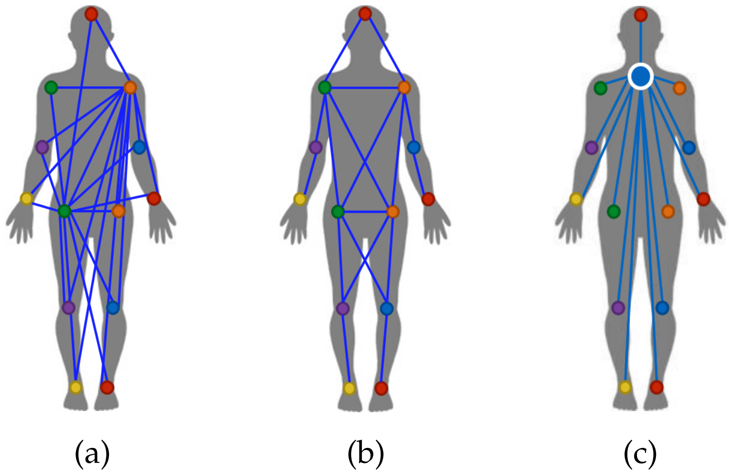


Figure 4.3: Visualization of (a) dense connectivity *BU-full* (for clarity only a subset of connections is shown, *e.g.* left hip and right shoulder are connected to all other body parts), (b) sparse connectivity *BU-sparse*, (c) star connectivity, where all body parts are connected only to the root node and not to each other.

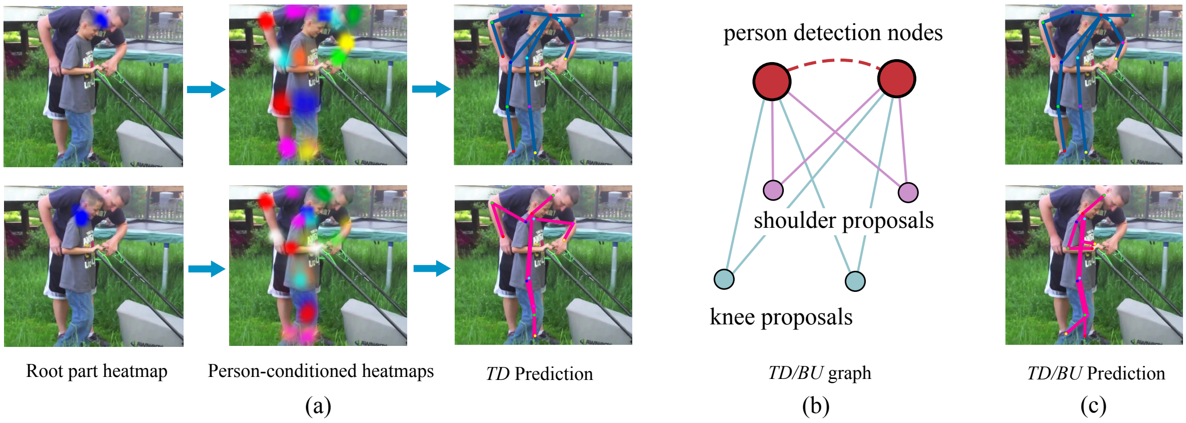


Figure 4.4: **(a)** Processing stages of the *Top-Down* model shown for an example with significantly overlapping people. Left: Heatmaps for the chin (=root part) used to condition the CNN on the location of the person in the back (top) and in the front (bottom). Middle: Output heatmaps for all body parts, notice the ambiguity in estimates of the arms of the front person. Right: *TD* predictions for each person. **(b)** Example of the *Top-Down/Bottom-Up* graph. Red dotted line represents the must-cut constraint. Note that body part proposals of different type are connected to person nodes but not between each other. **(c)** *Top-Down/Bottom-Up* predictions. Notice that the *TD/BU* inference correctly assigns the forearm joints of the frontal person.

cost d_e is given by the output of the logistic regression given the features computed from offset and angle of the predicted and actual location of the other joint in the pair. We refer to Insafutdinov *et al.* (2016a) for more details on these pairwise terms.

Note that our model generalizes the model of Tang *et al.* (2015) in that the edge cost depends on the type of nodes linked by the edge. It also generalizes two recent models (Pishchulin *et al.*, 2016; Insafutdinov *et al.*, 2016a) by allowing G to be sparse. This is achieved by reformulating the model with a more general type of cycle constraint (4.5), in contrast to simple triangle inequalities used in prior multi-person pose estimation work (Pishchulin *et al.*, 2016; Insafutdinov *et al.*, 2016a)[1].

4.3.2 Top-Down/Bottom-up Model (*TD/BU*)

We now introduce a version of our model that operates by first generating body part proposals conditioned on the locations of people in the image and then performing joint reasoning to group these proposals into spatio-temporal clusters corresponding to different people. We follow the intuition that it is considerably easier to identify and detect individual people (e.g. by detecting their heads) compared to correctly associating body parts such as ankles and wrists to each person. We select person's head as a root part that is responsible for representing the person location, and delegate the task of identifying body parts of the person corresponding to a head location to a convolutional network.

The structure of *TD/BU* model is illustrated in Figure 4.4 (b) for the simplified case of two distinct head detections. Let us denote the set of all root part detections as $D^{root} = \{d_i^{root}\}$. For each pair of the root nodes we explicitly set the corresponding edge indicator variables $y_{d_j^{root}, d_k^{root}} = 0$. This implements a "must-not-link" constraint between these nodes, and in combination with the cycle inequality (4.5) implies that each proposal can be connected to one of the "person nodes" only. The cost for an edge connecting detection proposal $\mathbf{d}_k$ and a "person node" d_i^{root} is based on the conditional distribution $p_{d_k^c}(d_k^{pos} | d_i^{root})$ generated by the convolutional network. The output of such network is a set of conditional distributions, one for each node type. We augment the graph G with attractive/repulsive and temporal terms as described in Section 4.3.4 and Section 4.3.3 and set the unary costs for all indicator variables x_d to a constant. Any proposal not connected to any of the root nodes is excluded from the final solution. We use the solver introduced by Levinkov *et al.* (2017) for consistency, but a simpler KL-based solver as in the recent multi-target tracking works (Tang *et al.*, 2015; Keuper *et al.*, 2015) could be used as well since the *TD/BU* model effectively ignores the unary variables x_d. The processing stages of *TD/BU* model are shown in Figure 4.4. Note that the body-part heatmaps change depending on the person-identity signal provided by the person's neck, and that the bottom-up step was able to correct the predictions on the forearms of the front person.

Implementation details. For head detection, we use a version of our model that

[1] See Section 2.1 in Pishchulin *et al.* (2016)

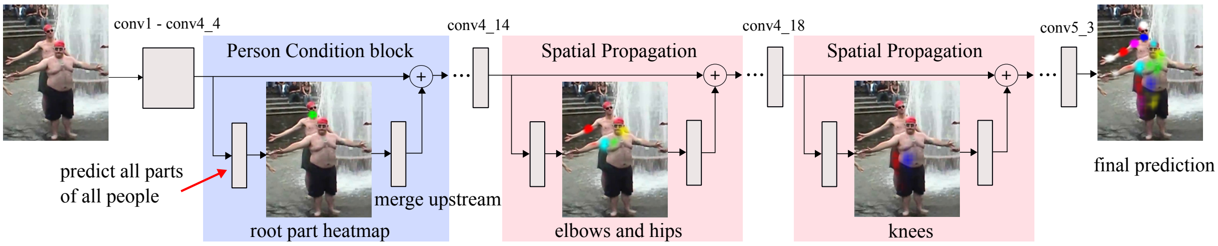


Figure 4.5: CNN architecture based on ResNet-101 for computing person conditioned proposals and pairwise terms. *SP* block for shoulders at *conv_4_8* is omitted for clarity.

contains the two head parts (neck and head top). This makes our *TD/BU* model related to the hierarchical model defined in the work of Insafutdinov *et al.* (2016a) that also uses easier-to-detect parts to guide the rest of the inference process. However here we replace all the stages in the hierarchical inference except the first one with a convolutional network.

The structure of the convolutional network used to generate person-conditioned proposals is shown on Figure 4.5. The network uses the ResNet-101 (He *et al.*, 2016) that we modify to bring the stride of the network down to 8 pixels (Insafutdinov *et al.*, 2016a). The network generates predictions for all body parts after the *conv4_4* block. We use the cross-entropy binary classification loss at this stage to predict the part heatmaps. At each training iteration we forward pass an image with multiple people potentially in close proximity to each other. We select a single person from the image and condition the network on the person's neck location by zeroing out the heatmap of the neck joint outside the ground-truth region. We then pass the neck heatmap through a convolutional layer to match the dimensionality of the feature channels and add them to the main stream of the ResNet. We finally add a joint prediction layer at the end of the network with a loss that considers predictions to be correct only if they correspond to the body joints of the selected person.

Spatial propagation (*SP*). In our network the person identity signal is provided by the location of the head. In principle the receptive field size of the network is large enough to propagate this signal to all body parts. However we found that it is useful to introduce an additional mechanism to propagate the person identity signal. To that end we inject intermediate supervision layers for individual body parts arranged in the order of kinematic proximity to the root joint (Figure 4.5). We place prediction layers for shoulders at *conv4_8*, for elbows and hips at *conv4_14* and for knees at *conv4_18*. We empirically found that such an explicit form of spatial propagation significantly improves performance on joints such as ankles, that are typically far from the head in the image space (see Table 4.2 for details).

Training. We use Caffe's (Jia *et al.*, 2014) ResNet implementation and initialize from the ImageNet-pre-trained models. Networks are trained on the MPII Human Pose dataset (Andriluka *et al.*, 2014) with SGD for 1M iterations with stepwise learning rate (lr=0.002 for 400k, lr=0.0002 for 300k and lr=0.0001 for 300k).

4.3.3 Temporal Model

Regardless of the type of within frame model (*BU* or *TD/BU*) we rely on the same type of temporal edges that connect nodes of the same type in adjacent frames. We derive the costs for such temporal edges via logistic regression. Given the feature vector g_{ij} the probability that the two proposals $\mathbf{d}_i$ and $\mathbf{d}_j$ in adjacent frames correspond to the same body part is given by: $p(y_{ij} = 1|g_{ij}) = 1/(1 + \exp(-\langle \omega_t, g_{ij} \rangle))$, where $g_{ij} = (\Delta_{ij}^{L2}, \Delta_{ij}^{Sift}, \Delta_{ij}^{DM}, \tilde{\Delta}_{ij}^{DM})$, and $\Delta_{ij}^{L2} = \|d_i^{pos}$ - $d_j^{pos}\|_2$, Δ_{ij}^{Sift} is Euclidean distance between the SIFT descriptors computed at d_i^{pos} and d_j^{pos}, and Δ_{ij}^{DM} and $\tilde{\Delta}_{ij}^{DM}$ measure the agreement with the dense motion field computed with the DeepMatching approach of Weinzaepfel *et al.* (2013).

For SIFT features we specify the location of the detection proposal, but rely on SIFT to identify the local orientation. In cases with multiple local maxima in orientation estimation we compute SIFT descriptor for each orientation and set Δ_{ij}^{Sift} to the minimal distance among all pairs of descriptors. We found that this makes the SIFT distance more robust in the presence of rotations of the body limbs.

We define the features Δ_{ij}^{DM} and $\tilde{\Delta}_{ij}^{DM}$ as in the work of Tang *et al.* (2016). Let $R_i = R(\mathbf{d}_i)$ be an squared image region centered on the part proposal $\mathbf{d}_i$. We define Δ_{ij}^{DM} as a ratio of the number of point correspondences between the regions R_i and R_j and the total number of point correspondences in either of them. Specifically, let $C = \{c^k | k = 1, \ldots, K\}$ be a set of point correspondences between the two images computed with DeepMatching, where $c^k = (c_1^k, c_2^k)$ and c_1^k and c_2^k denote the corresponding points in the first and second image respectively. Using this notation we define:

$$\Delta_{ij}^{DM} = \frac{|\{c_k | c_1^k \in R_i \wedge c_2^k \in R_j\}|}{|\{c_k | c_1^k \in R_i\}| + |\{c_k | c_2^k \in R_j\}|}. \tag{4.6}$$

The rationale behind computing Δ_{ij}^{DM} by aggregating across multiple correspondences is to make the feature robust to outliers and to inaccuracies in body part detection. $\tilde{\Delta}_{ij}^{DM}$ is defined analogously, but using the DeepMatching correspondences obtained by inverting the order of images.

Discussion. As we demonstrate in Section 4.4, we found the set of features described above to be complementary to each other. Euclidean distance between proposals is informative for finding correspondences for slow motions, but fails for faster motions and in the presence of multiple people. DeepMatching is usually effective in finding corresponding regions between the two images, but occasionally fails in the case of sudden background changes due to fast motion or large changes in body limb orientation. In these cases SIFT is often still able to provide a meaningful measure of similarity due to its rotation invariance.

4.3.4 Attractive/Repulsive Edges

In addition to the *cross-type* and *temporal* edges the *BU* and *TD/BU* models described above include attractive/repulsive edges that are defined following the model in Chapter 3. These edges connect each pair of proposals of the same type within the same image and have the costs that is inversely-proportional to the distance between the proposal locations. The inclusion of attractive/repulsive edges leads to an effect similar to non-maximum suppression but the decision to suppress a proposal is made based on the evidence from the entire image. This is in contrast to typical non-maximum suppression based on the detection scores of two proposals only. Another function of attractive/repulsive edges is to prevent grouping of multiple distant hypothesis of the same type, *e.g.* prevent grouping of the heads of two different people.

4.4 EXPERIMENTS

4.4.1 Datasets and Evaluation Measure

Single frame. We evaluate our single frame models on the MPII Multi-Person dataset (Andriluka *et al.*, 2014). We report all intermediate results on a validation set of 200 images sampled uniformly at random (MPII Multi-Person Val), while major results and comparison to the state of the art are reported on the test set.

Video. In order to evaluate video-based models we introduce a novel "MPII Video Pose" dataset[2] that is composed of short image sequences around keyframes from the MPII Multi-Person. Selected keyframes represent crowded scenes with highly articulated people engaging in various dynamic activities. In addition to each keyframe, we include +/-10 neighboring frames from the corresponding publicly available video sequences, and annotate every second frame[3]. Each body pose was annotated following the standard annotation procedure (Andriluka *et al.*, 2014), while maintaining person identity throughout the sequence. In contrast to MPII Multi-Person where some frames may contain non-annotated people, we annotate all people participating in the activity captured in the video, and add ignore regions for areas that contain dense crowds (e.g. static spectators in the dancing sequences). In total, our dataset consists of 28 sequences with over $2,000$ annotated poses.

Finally, to enable comparison with the state-of-the-art pose tracking methods, we present the results of our model on the recently proposed PoseTrack dataset (Andriluka *et al.*, 2018). We introduce two simplifications that follow the work by Papandreou *et al.* (2017). First, we rely on a person detector to establish locations of people in the image and run pose estimation independently for each person detection. This allows us to deal with large variation in scale present in the dataset by cropping and rescaling images to canonical scale prior to pose estimation. In

[2] Dataset is available at pose.mpi-inf.mpg.de/art-track.

[3] The annotations in the original key-frame are kept unchanged.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	τ_{CNN}	τ_{graph}
BU-full, label	90.0	84.9	71.1	58.4	69.7	64.7	54.7	70.5	**0.18**	3.06
BU-full	91.2	86.0	**72.9**	61.5	70.4	65.4	55.5	71.9	**0.18**	0.38
BU-sparse	91.1	**86.5**	70.7	58.1	69.7	64.7	53.8	70.6	**0.18**	0.22
TD/BU + SP	**92.2**	86.1	72.8	**63.0**	**74.0**	**66.2**	**58.4**	**73.3**	0.94[7]	**0.08**

Table 4.1: Effects of various variants of *BU* model on pose estimation performance (AP) on MPII Multi-Person Val and comparison to the best variant of *TD/BU*model.

addition, this also allows us to group together the body-part estimates inferred for a given detection bounding box. As a second simplification we apply the model on the level of full body poses and not on the level of individual body parts. We use a publicly available Faster-RCNN (Ren *et al.*, 2015) detector from the TensorFlow Object Detection API (Huang *et al.*, 2016) for people detection. This detector has been trained on the "MS COCO" dataset and uses Inception-ResNet-V2 (Szegedy *et al.*, 2017) for image encoding. We refer to this simplified model as *TD/Simple* throughout the experiments.

Evaluation details. The average precision (AP) measure (Pishchulin *et al.*, 2016) is used for evaluation of pose estimation accuracy. For each algorithm we also report run time τ_{CNN} of the proposal generation and τ_{graph} of the graph partitioning stages. All time measurements were conducted on a single core Intel Xeon 2.70GHz. Finally we also evaluate tracking perfomance using standard MOTA metric (Bernardin and Stiefelhagen, 2008).

Evaluation on our "MPII Video Pose" dataset is performed on the full frames using the publicly available evaluation kit by Andriluka *et al.* (2014). On MPII Multi-Person we follow the official evaluation protocol[4] and evaluate on groups using the provided rough group location and scale.

4.4.2 Single-frame Models

We compare the performance of different variants of our *Bottom-Up* (*BU*) and *Top-Down/Bottom-Up* (*TD/BU*) models introduced in Section 4.3.1 and Section 4.3.2. For *BU* we consider a model that (1) uses a fully-connected graph with up to 1,000 detection proposals and jointly performs partitioning *and* body-part labeling similar to the model in Chapter 3 (*BU-full, label*); (2) is same as (1), but labeling of detection proposals is done based on detection score (*BU-full*); (3) is same as (2), but uses a sparsely-connected graph (*BU-sparse*). The results are shown in Table 4.1[5]. *BU-full*,

[4] http://human-pose.mpi-inf.mpg.de/#evaluation

[5] Our current implementation of *TD/BU* operates on the whole image when computing person-conditioned proposals and computes the proposals sequentially for each person. More efficient implementation would only compute the proposals for a region surrounding the person and run

label achieves 70.5% AP with a median inference run-time τ_{graph} of 3.06 s/f. *BU-full* achieves 8× run-time reduction (0.38 vs. 3.06 s/f): pre-labeling detection candidates based on detection score significantly reduces the number of variables in the problem graph. Interestingly, pre-labeling also improves the performance (71.9 vs. 70.5% AP): some of the low-scoring detections may complicate the search for an optimal labeling. *BU-sparse* further reduces run-time (0.22 vs. 0.38 s/f), as it reduces the complexity of the initial problem by sparsifying the graph, at a price of a drop in performance (70.6 vs. 71.9% AP).

In Table 4.2 we compare the variants of the *TD/BU* model. Our *TD* approach achieves 71.7% AP, performing on par with a more complex *BU-full*. Explicit spatial propagation (*TD+SP*) further improves the results (72.5 vs. 71.7% AP). The largest improvement is observed for ankles: progressive prediction that conditions on the close-by parts in the tree hierarchy reduces the distance between the conditioning signal and the location of the predicted body part and simplifies the prediction task. Performing inference (*TD/BU+SP*) improves the performance to 73.3% AP, due to more optimal assignment of part detection candidates to corresponding persons. Graph simplification in *TD/BU* allows to further reduce the inference time for graph partitioning (0.08 vs. 0.22 for *BU-sparse*).

Qualitative results. We perform qualitative comparison of the proposed single-frame based *TD/BU* and *BU-full* methods on challenging scenes containing highly articulated and strongly overlapping individuals. Results are shown in Figure 4.6 and Figure 4.7. We observe that the *BU-full* tends to fail on images where people significantly overlap (images 1-3, 5-10) or exhibit high degree of articulation (image 4). This is due to the fact that geometric image-conditioned pairwise may get confused in the presence of multiple overlapping individuals and thus mislead post-CNN bottom-up assembling of body poses. In contrast, *TD/BU* performs explicit modeling of person identity via top-dop bottom-up reasoning while offloading the larger share of the reasoning about body-part association onto feed-forward convolutional architecture, and thus is able to resolve such challenging cases. Interestingly, *TD/BU* is able to correctly predict lower limbs of people in the back through partial occlusion (image 3, 5, 7, 10).

Comparison to the State of the Art. We compare the proposed single-frame approaches to the state of the art on MPII Multi-Person Test and WAF (Eichner and Ferrari, 2010) datasets. Comparison on MPII is shown in Table 4.3. Both *BU-full* and *TD/BU* improve over the best published result of *DeeperCut* (Insafutdinov *et al.*, 2016b), achieving 72.9 and 74.3% AP respectively vs. 70.0% AP by *DeeperCut*. For the *TD/BU* the improvements on articulated parts (elbows, wrists, ankles, knees) are particularly pronounced. We argue that this is due to using the network that is directly trained to disambiguate body parts of different people, instead of using explicit

multiple people in a single batch. Clearly in cases when two people are close in the image this would still process the same image region multiple times. However the image regions far from any person would be excluded from processing entirely. On average we expect similar image area to be processed during proposal generation stage in both *TD/BU* and *BU-sparse*, and expect the runtimes τ_{CNN} to be comparable for both models.

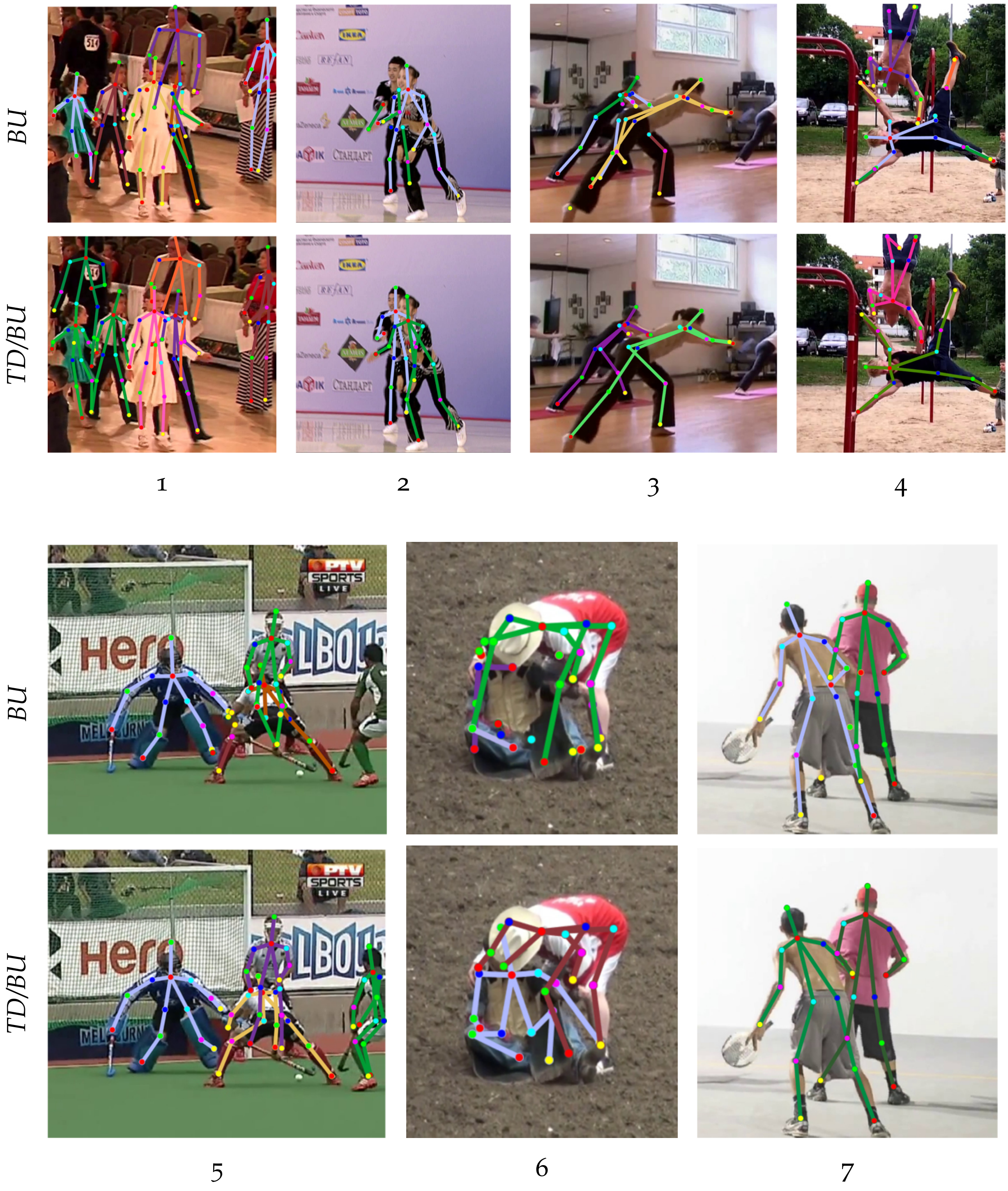


Figure 4.6: Qualitative comparison of single-frame based *TD/BU* and *BU-full* on MPII Multi-Person dataset.

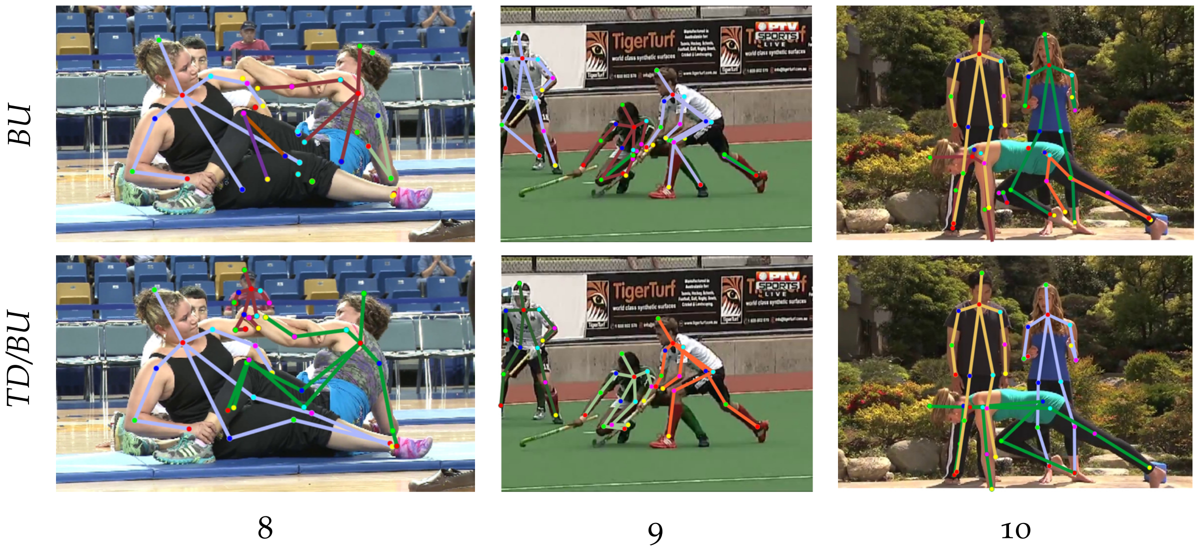


Figure 4.7: Qualitative comparison of single-frame based *TD/BU* and *BU-full* on MPII Multi-Person dataset.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP
TD	91.6	84.7	**72.9**	**63.2**	72.3	64.7	52.8	71.7
TD + SP	90.7	85.0	72.0	63.1	73.1	65.0	58.3	72.5
TD/BU + SP	**92.2**	**86.1**	72.8	63.0	**74.0**	**66.2**	**58.4**	**73.3**

Table 4.2: Effects of various versions of *TD/BU* model on pose estimation performance (AP) on MPII Multi-Person Val.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP	τ_{graph}
BU-full	**91.5**	**87.8**	74.6	62.5	72.2	65.3	56.7	72.9	0.12
TD/BU+ SP	88.8	87.0	**75.9**	**64.9**	**74.2**	**68.8**	**60.5**	**74.3**	**0.005**
DeeperCut Insafutdinov *et al.* (2016a)	79.1	72.2	59.7	50.0	56.0	51.0	44.6	59.4	485
DeeperCut Insafutdinov *et al.* (2016b)	89.4	84.5	70.4	59.3	68.9	62.7	54.6	70.0	485
Iqbal and Gall (2016)	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10

Table 4.3: Pose estimation results (AP) on MPII Multi-Person Test.

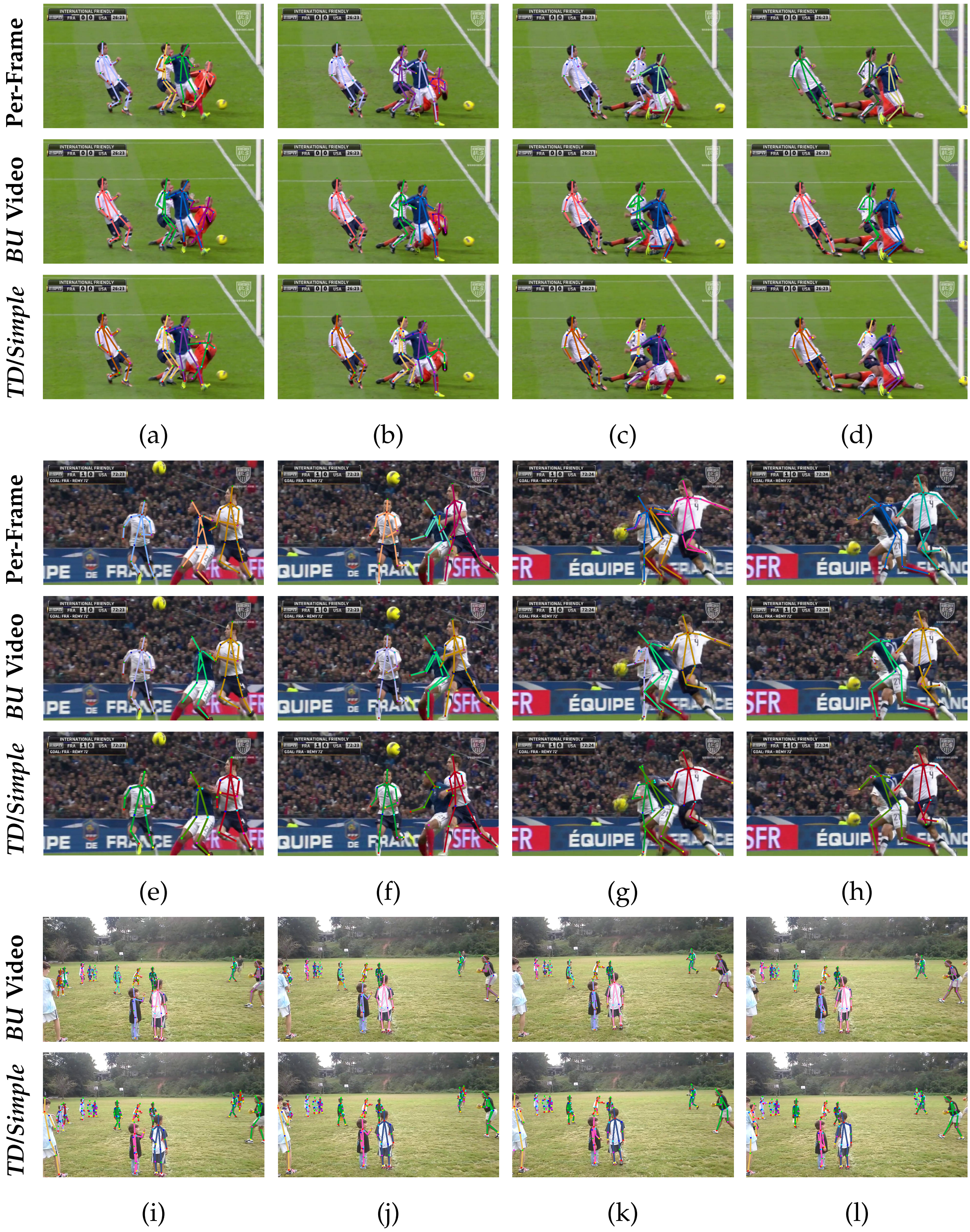


Figure 4.8: Qualitative comparison of results using single frame based model (*BU-sparse*) vs. articulated tracking (*BU-sparse+temporal*) vs. simlified *TD/Simple* model. See http://youtube.com/watch?v=eYtn13fzGGo for the supplemental material showcasing our results.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP
BU-full	84.0	83.8	73.0	61.3	74.3	67.5	58.8	71.8
+ *temporal*	84.9	83.7	72.6	61.6	74.3	68.3	59.8	72.2
BU-sparse	84.5	84.0	71.8	59.5	**74.4**	68.1	59.2	71.6
+ *temporal*	**85.6**	84.5	73.4	62.1	73.9	68.9	63.1	73.1
TD/BU+ SP	82.2	85.0	75.7	64.6	74.0	69.8	62.9	73.5
+ *temporal*	82.6	**85.1**	**76.3**	**65.5**	74.1	**70.7**	**64.7**	**74.2**

Table 4.4: Pose estimation results (AP) on "MPII Video Pose".

geometric pairwise terms that only serve as a proxy to person's identity. Overall, the performance of our best *TD/BU* method is noticeably higher (74.3 vs. 70.0% AP). Remarkably, its run-time τ_{graph} of graph partitioning stage is 5 orders of magnitude faster compared to *DeeperCut*. This speed-up is due to two factors. First, *TD/BU* relies on a faster solver (Levinkov *et al.*, 2017) that tackles the graph-partitioning problem via local search, in contrast to the exact solver used by Insafutdinov *et al.* (2016a). Second, in the case of *TD/BU* model the graph is sparse and a large portion of the computation is performed by the feed-forward CNN introduced in Section 4.3.2. On WAF (Eichner and Ferrari, 2010) dataset *TD/BU* substantially improves over the best published result (87.7 vs. 82.0% AP by Insafutdinov *et al.* (2016b)).

4.4.3 Multi-frame Models

Comparison of video-based models. Performance of the proposed video-based models is compared in Table 4.4. Video-based models outperform single-frame models in each case. *BU-full+temporal* slightly outperforms *BU-full*, where improvements are noticeable for ankle, knee and head. *BU-sparse+temporal* noticeably improves over *BU-sparse* (73.1 vs. 71.6% AP). We observe significant improvements on the most difficult parts such as ankles (+3.9% AP) and wrists (+2.6% AP). Interestingly, *BU-sparse+temporal* outperforms *BU-full + temporal*: longer-range connections such as, *e.g.* , head to ankle, may introduce additional confusion when information is propagated over time. Finally, *TD/BU+temporal* improves over *TD/BU* (+0.7% AP). Similarly to *BU-sparse+temporal*, improvement is most prominent on ankles (+1.8% AP) and wrists (+0.9% AP). Note that even the single-frame *TD/BU* outperforms the best temporal *BU* model. We show examples of articulated tracking on "MPII Video Pose" in Figure 4.8. Temporal reasoning helps in cases when image information is ambiguous due to close proximity of multiple people. For example the video-based approach succeeds in correctly localizing legs of the person in Figure 4.8 (d) and (h).

Temporal features. We evaluate the importance of combining temporal features introduced in Section 4.3.3 on our Multi-Person Video dataset. To that end, we

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	AP
BU-sparse	84.5	84.0	71.8	59.5	74.4	68.1	59.2	71.6
+ *det-distance*	84.8	84.3	72.9	61.8	74.1	67.4	59.1	72.1
+ *deepmatch*	85.5	83.9	73.0	62.0	74.0	68.0	59.5	72.3
+ *det-distance*	85.1	83.6	72.2	61.5	**74.4**	68.8	62.2	72.5
+ *sift-distance*	**85.6**	**84.5**	**73.4**	**62.1**	73.9	**68.9**	**63.1**	**73.1**

Table 4.5: Effects of different temporal features on pose estimation performance (AP) (*BU-sparse+temporal* model) on our "MPII Video Pose".

consider *BU-sparse+temporal* model and compare results to *BU-sparse* in Table 4.5. Single-frame *BU-sparse* achieves 71.6% AP. It can be seen that using geometry based *det-distance* features slightly improves the results to 72.1% AP, as it enables the propagation of information from neighboring frames. Using *deepmatch* features slightly improves the performance further as it helps to link the same body part of the same person over time based on the body part appearance. It is especially helpful in the case of fast motion where *det-distance* may fail. The combination of both geometry and appearance based features further improves the performance to 72.5%, which shows their complementarity. Finally, adding the *sift-distance* feature improves the results to 73.1%, since it copes better with the sudden changes in background and body part orientations. Overall, using a combination of temporal features in *BU-sparse+temporal* results in a 1.5% AP improvement over the single-frame *BU-sparse*. This demonstrates the advantages of the proposed approach to improve pose estimation performance using temporal information.

Tracking evaluation. In Table 4.6 we present results of the evaluation of multi-person articulated body tracking. We treat each body joint of each person as a tracking target and measure tracking performance using a standard multiple object tracking accuracy (MOTA) metric (Bernardin and Stiefelhagen, 2008) that incorporates identity switches, false positives and false negatives[6]. We experimentally compare to a baseline model that first tracks people across frames and then performs per-frame pose estimation. To track a person we use a reduced version of our algorithm that operates on the two head joints only. This allows to achieve near perfect person tracking results in most cases. Our tracker still fails when the person head is occluded for multiple frames as it does not incorporate long-range connectivity between target hypothesis. We leave handling of long-term occlusions for the future work. For full-body tracking we use the same inital head tracks and add them to the set of body part proposals, while also adding must-link and must-cut constraints for the temporal edges corresponding to the head parts detections. The rest of the

[6] Note that MOTA metric does not take the confidence scores of detection or track hypotheses into account. To compensate for that in the experiment in Table 4.6 we remove all body part detections with a score ≤ 0.65 for *BU-sparse* and ≤ 0.7 for *TD/BU* prior to evaluation.

Setting	Head	Sho	Elb	Wri	Hip	Knee	Ank	Average
Head track + *BU-sparse*	70.5	71.7	53.0	41.7	57.0	52.4	41.9	55.5
+ *temporal*	**70.6**	**72.7**	**58.0**	47.6	57.6	54.8	47.7	**58.5**
Head track + *TD/BU*	64.8	69.4	55.4	43.4	56.4	52.2	44.8	55.2
+ *temporal*	65.0	69.9	56.3	44.2	56.7	53.2	46.1	55.9
TD/Simple	65.5	69.8	57.5	**47.9**	**57.8**	**58.7**	**52.2**	**58.5**

Table 4.6: Tracking results (MOTA) on the "MPII Video Pose".

graph remains unchanged so that at inference time the body parts can be freely assigned to different person tracks. For the *BU-sparse* the full body tracking improves performance by +5.9 and +5.8 MOTA on wrists and ankles, and by +5.0 and +2.4 MOTA on elbows and knees respectively. *TD/BU* benefits from adding temporal connections between body parts as well, but to a lesser extent than *BU-sparse*. *BU-sparse* also achieves the best overall score of 58.5 compared to 55.9 by *TD/BU*. This is surprising since *TD/BU* outperformed *BU-sparse* on the pose estimation task (see Table 4.1 and 4.3). We hypothesize that limited improvement of *TD/BU* could be due to balancing issues between the temporal and spatial pairwise terms that are estimated independently of each other.

Finally, we present results of our model on the PoseTrack benchmark in Table 4.7 and compare them to another graph partitioning approach by Iqbal *et al.* (2017b). Both methods perform on par, with our's demonstrating better tracking of terminal joints such as wrists and ankles, while Posetrack shows better accuracy on head and shoulders. We also compare the simplified *TD/Simple* model with our full models (*BU*and *TD/BU*) both quantitatively (Table4.6) and qualitatively (Figure 4.8). Even though the tracking accuracy of the models are comparable (both *BU-sparse* and *TD/Simple* achieve 58.5 MOTA), joint inference on a body-part level in *BU* allows it to better handle difficult crowded scenes, such as Figure 4.8 (b, c, f), where *TD/Simple* fails to assign parts of the lower body to the subjects correctly. On the other hand, we found that *TD/Simple* better handles cases with severe variations of scale such as the farthest person from the camera in the right of Figure 4.8 (i, j, l), which *BU-sparse* fails to detect completely.

Method	MOTA								AP
	Head	Sho	Elb	Wri	Hip	Knee	Ank	Total	Total
ours, *TD/Simple*	58.6	56.8	**47.9**	**41.0**	**47.6**	**45.2**	**39.6**	48.1	**59.4**
Iqbal *et al.* (2017b)	**59.3**	**64.9**	46.9	38.2	45.6	43.1	35.1	**48.4**	59.2

Table 4.7: Pose tracking results (MOTA) and pose estimation results (AP) on the PoseTrack'17 benchmark (Andriluka *et al.*, 2018).

4.5 CONCLUSION

In this chapter we introduced an efficient and effective approach to articulated body tracking in monocular video. Our approach defines a model that jointly groups body part proposals within each video frame and across time. Grouping is formulated as a graph partitioning problem that lends itself to efficient inference with recent local search techniques. Our approach improves over state-of-the-art while being substantially faster compared to other related work.

5 A BENCHMARK FOR HUMAN POSE ESTIMATION AND TRACKING

Contents

Existing systems for video-based pose estimation and tracking struggle to perform well on realistic videos with multiple people and often fail to output body-pose trajectories consistent over time. To address this shortcoming this chapter introduces PoseTrack which is a new large-scale benchmark for video-based human pose estimation and articulated tracking. Our new benchmark encompasses three tasks focusing on i) single-frame multi-person pose estimation, ii) multi-person pose estimation in videos, and iii) multi-person articulated tracking. To establish the benchmark, we collect, annotate and release a new dataset that features videos with multiple people labeled with person tracks and articulated pose. A public centralized evaluation server is provided to allow the research community to evaluate on a held-out test set. Furthermore, we conduct an extensive experimental study on recent approaches to articulated pose tracking and provide analysis of the strengths and weaknesses of the state of the art. We envision that the proposed benchmark will stimulate productive research both by providing a large and representative training dataset as well as providing a platform to objectively evaluate and compare the proposed methods. The benchmark is freely accessible at https://posetrack.net/.

5.1 INTRODUCTION

Human pose estimation has made significant progress on the tasks of single person pose estimation in individual frames (Toshev and Szegedy, 2014; Tompson *et al.*, 2014, 2015; Carreira *et al.*, 2016; Wei *et al.*, 2016; Hu and Ramanan, 2016; Insafutdinov *et al.*, 2016a; Newell *et al.*, 2016; Bulat and Tzimiropoulos, 2016; Rafi *et al.*, 2016) and videos (Pfister *et al.*, 2015; Charles *et al.*, 2016; Iqbal *et al.*, 2017a; Gkioxari *et al.*, 2016) as well as multi-person pose estimation in monocular images (Pishchulin *et al.*, 2016; Insafutdinov *et al.*, 2016a; Iqbal and Gall, 2016; Cao *et al.*, 2017; Papandreou *et al.*, 2017). This progress has been facilitated by the use of deep learning-based architectures (Simonyan and Zisserman, 2014; He *et al.*, 2016) and by the availability of large-scale benchmark datasets such as "MPII Human Pose" (Andriluka *et al.*, 2014) and "MS COCO" (Lin *et al.*, 2014). Importantly, these benchmark datasets not only have provided extensive training sets required for training of deep learning based approaches, but also established detailed metrics for direct and fair performance comparison across numerous competing approaches.

Despite significant progress of single frame based multi-person pose estimation, the problem of *articulated multi-person body joint tracking* in monocular video remains largely unaddressed. Although there exist training sets for special scenarios, such as sports (Zhang *et al.*, 2013; Jhuang *et al.*, 2013) and upright frontal people (Charles *et al.*, 2016), these benchmarks focus on *single isolated individuals* and are still limited in their scope and variability of represented activities and body motions. In this work, we aim to fill this gap by establishing a new large-scale, high-quality benchmark for video-based multi-person pose estimation and articulated tracking.

Our benchmark is organized around three related tasks focusing on single-frame multi-person pose estimation, multi-person pose estimation in video, and multi-person articulated tracking. While the main focus of the dataset is on multi-person articulated tracking, progress in the single-frame setting will inevitably improve overall tracking quality. We thus make the single frame multi-person setting part of our evaluation procedure. In order to enable timely and scalable evaluation on the held-out test set, we provide a centralized evaluation server. We strongly believe that the proposed benchmark will prove highly useful to drive the research forward by focusing on remaining limitations of the state of the art.

To sample the initial interest of the computer vision community and to obtain early feedback we have organized workshops and a competitions at ICCV'17[1] and ECCV'18[2]. We obtained largely positive feedback from the teams that participated in the competitions. We incorporate some of this feedback into this report. In addition we analyze the currently best performing approaches and highlight the common difficulties for pose estimation and articulated tracking.

[1] https://posetrack.net/workshops/iccv2017/

[2] https://posetrack.net/workshops/eccv2018/

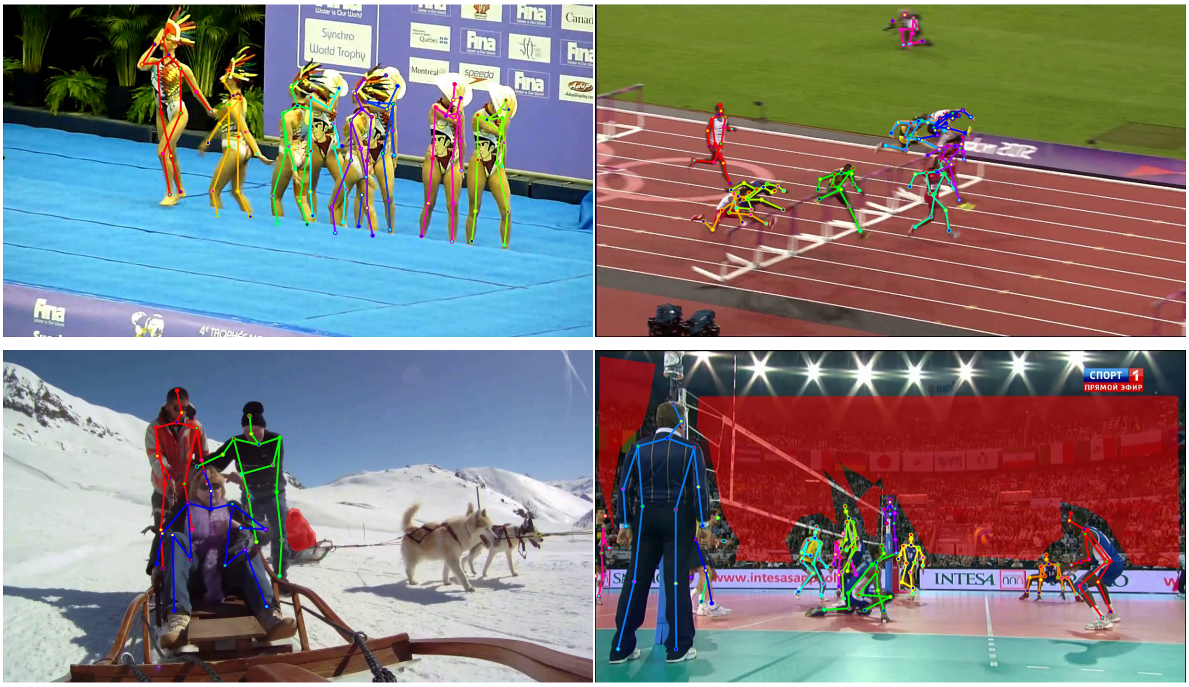


Figure 5.1: Example frames and annotations from our dataset.

5.2 THE POSETRACK DATASET AND CHALLENGE

We will now provide the details on data collection and the annotation process, as well as the established evaluation procedure. We build on and extend the newly introduced datasets for pose tracking in the wild (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). To that end, we use the raw videos provided by the popular MPII Human Pose dataset. For each frame in MPII Human Pose dataset we include $41 - 298$ neighboring frames from the corresponding raw videos, and then select sequences that represent crowded scenes with multiple articulated people engaging in various dynamic activities. The video sequences are chosen such that they contain a large amount of body motion and body pose and appearance variations. They also contain severe body part occlusion and truncation, *i.e.*, due to occlusions with other people or objects, persons often disappear partially or completely and re-appear again. The scale of the persons also varies across the video due to the movement of persons and/or camera zooming. Therefore, the number of visible persons and body parts also varies across the video.

5.2.1 Data Annotation

We annotated the selected video sequences with person locations, identities, body pose and ignore regions. The annotations were performed in four steps. First, we labeled ignore regions to exclude crowds and people for which pose can not be reliably determined due to poor visibility. Afterwards, the head bounding boxes for each person across the videos were annotated and a track ID was assigned to each person. The head bounding boxes provide an estimate of the absolute scale of the person required for evaluation. We assign a unique track ID to each person appearing in the video until the person moves out of the camera field-of-view. Note that each video in our dataset might contain several shots. We do not maintain track ID between shots and same person might get different track ID if it reappears in another shot. Poses for each person track are then annotated in the entire video. We annotate 15 body parts for each body pose including *head, nose, neck, shoulders, elbows, wrists, hips, knees and ankles*. All pose annotations were performed using the VATIC tool (Vondrick *et al.*, 2012) that allows to speed-up annotation by interpolating between frames. We chose to skip annotation of the body joints that can not be reliably localized by the annotator due to strong occlusion or difficult imaging conditions. This has proven the be a faster alternative to requiring annotators to guess the location of the joint and/or marking it as occluded. Figure 5.1 shows example frames from the dataset. Note the variability in appearance and scale, and complexity due to substantial number of people in close proximity.

The initial version of the dataset that we collected for the ICCV'17 workshop contained 550 video sequences with 66,374 frames. We split them into 292, 50, 208 videos for training, validation and testing, respectively. The split follows the original split of the MPII Human Pose dataset making it possible to train a model on the MPII Human Pose and evaluate on our test and validation sets.

The length of the majority of the sequences in our dataset ranges between 41 and 151 frames. The sequences correspond to about five seconds of video. Differences in the sequence length are due to variation in the frame rate of the videos. A few sequences in our dataset are longer than five seconds with the longest sequence having 298 frames. For each sequence in our benchmark we annotate the 30 frames in the middle of the sequence. In addition, we densely annotate validation and test sequences with a step of four frames. The rationale behind this annotation strategy is that we aim to evaluate both smoothness of body joint tracks as well as ability to track body joints over longer number of frames. We did not densely annotate the training set to save the annotation resources for the annotation of the test and validation set.

The ICCV'17 version of the dataset (we will refer to it as PoseTrack 2017) contained around 23,000 labeled frames with 153,615 pose annotations. In 2018 we undertook a second annotation effort and presented the larger dataset at the ECCV'18 workshop and competition. PoseTrack 2018 contains 1138 video sequences split into 593, 173, 375 videos for training, validation and testing respectively. In total, we provide almost 47,000 labeled frames with 276,198 pose annotations. To the best of our knowledge this makes PoseTrack the largest multi-person pose estimation and tracking dataset released to date. In Figure 5.2 we show additional statistics of the validation and test sets of our dataset. The plots show the distributions of the number of people per frame and per video, the track length and people sizes measured by the head bounding box. Note that substantial portion of the videos has a large number of people as shown in the plot on the top-right. The abrupt fall off in

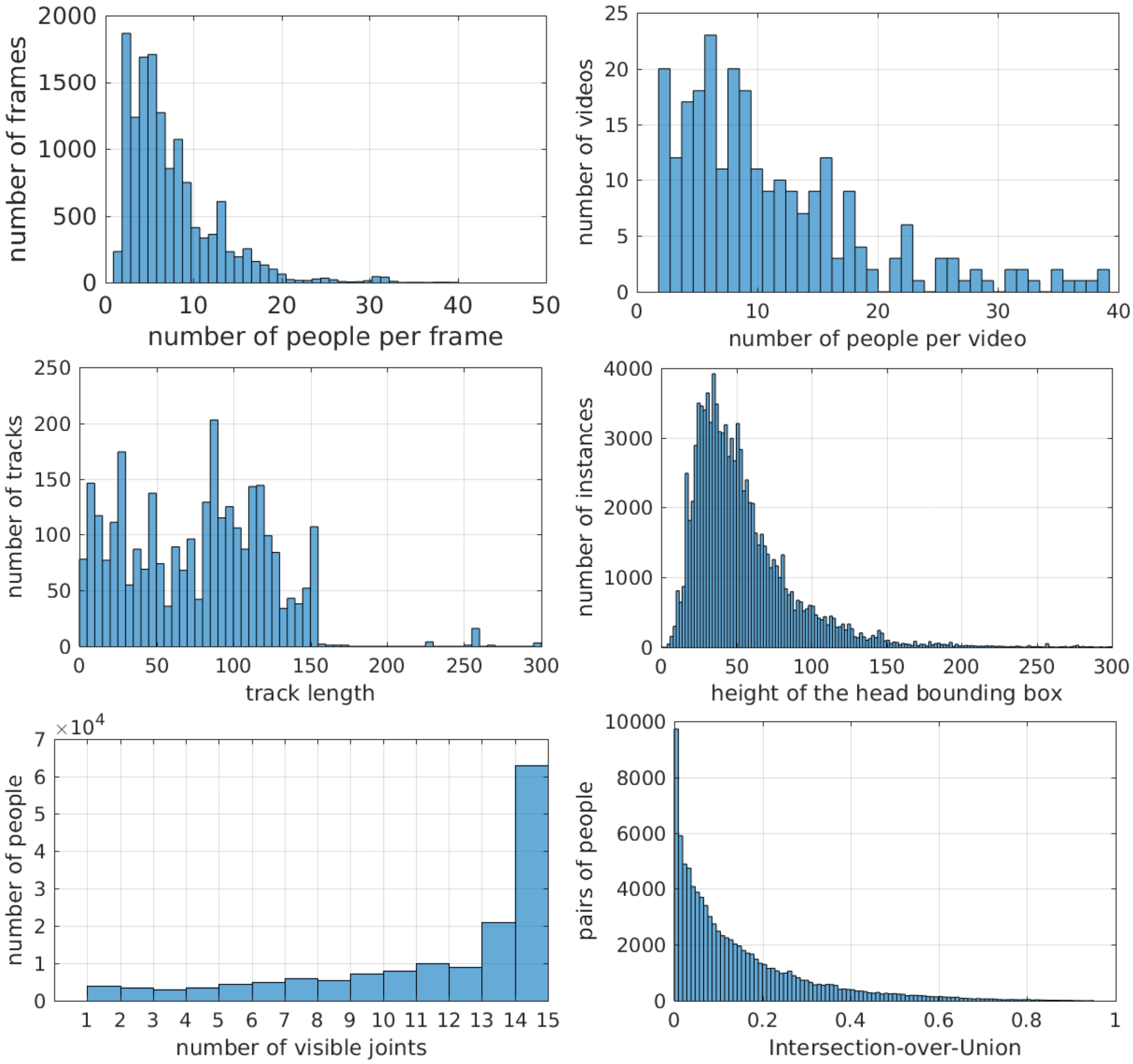


Figure 5.2: Various statistics of the PoseTrack benchmark.

the plot of the track length in the bottom-left is due to fixed length of the sequences included in the dataset.

5.2.2 Challenges

The benchmark consists of the following challenges:

Single-frame pose estimation. This task is similar to the ones covered by existing datasets like MPII Pose and MS COCO Keypoints, but on our new large-scale dataset.

Pose estimation in videos. The evaluation of this challenge is performed on single frames, however, the data will also include video frames before and after the annotated ones, allowing methods to exploit video information for a more robust single-frame pose estimation.

Pose tracking. This task requires to provide temporally consistent poses for all people visible in the videos. Our evaluation include both individual pose accuracy as well as temporal consistency measured by identity switches.

5.2.3 Evaluation Server

We provide an online evaluation server to quantify the performance of different methods on the held-out test set. This will not only prevent over-fitting to the test data but also ensures that all methods are evaluated in the exact same way, using the same ground truth and evaluation scripts, making the quantitative comparison meaningful. Additionally, it can also serve as a central directory of all available results and methods.

5.2.4 Experimental Setup and Evaluation Metrics

Since we need to evaluate both the accuracy of multi-person pose estimation in individual frames and articulated tracking in videos, we follow the best practices followed in both multi-person pose estimation (Pishchulin *et al.*, 2016) and multi-target tracking (Milan *et al.*, 2016). In order to evaluate whether a body part is predicted correctly, we use the PCKh (head-normalized probability of correct keypoint) metric (Andriluka *et al.*, 2014), which considers a body joint to be correctly localized if the predicted location of the joint is within a certain threshold from the true location. Due to large scale variation of people across videos and even within a frame, this threshold needs to be selected adaptively based on the person's size. To that end, we follow the work of Andriluka *et al.* (2014) and use 50% of the head length where the head length corresponds to 60% of the diagonal length of the ground-truth head bounding box. Given the joint localization threshold for each person, we compute two sets of evaluation metrics, one which is commonly used for evaluating multi-person pose estimation (Pishchulin *et al.*, 2016), and one from the multi-target tracking literature (Yang and Nevatia, 2012; Choi, 2015; Milan *et al.*, 2016) to evaluate multi-person pose tracking. During evaluation we ignore all person detections that overlap with the ignore regions.

Multi-person pose estimation. For measuring frame-wise multi-person pose accuracy, we use *mean Average Precision* (mAP) as is done by Pishchulin *et al.* (2016). The protocol to evaluate multi-person pose estimation by Pishchulin *et al.* (2016) requires that the location of a group of persons and their rough scale is known during evaluation. This information, however, is almost never available in realistic scenarios, particularly for videos. We therefore, propose not to use any ground-truth information during testing and evaluate the predictions without rescaling or selecting a specific group of people for evaluation.

Articulated multi-person pose tracking. To evaluate multi-person pose tracking, we use Multiple Object Tracking (MOT) metrics (Milan *et al.*, 2016) and apply them independently to each of the body joints. Metrics measuring the overall tracking performance are then obtained by averarging the per-joint metrics. The metrics require predicted body poses with track IDs. First, for each frame, for each body joint class, distances between predicted and ground-truth locations are computed. Subsequently predicted and ground-truth locations are matched to each other by a global matching procedure that minimizes the total assignment distance. Finally, Multiple Object Tracker Accuracy (MOTA), Multiple Object Tracker Precision (MOTP), Precision, and Recall metrics are computed. Evaluation server reports MOTA metric for each body joint class and average over all body joints, while for MOTP, Precision, and Recall we report averages only. In the following evaluation MOTA is used as our main tracking metric. The source code for the evaluation metrics is publicly available on the benchmark website.

5.3 analysis of the state of the art

Articulated pose tracking in unconstrained videos is a relatively new topic in computer vision research. To the best of our knowledge only few approaches for this task have been proposed in the literature (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). Therefore, to analyse the performance of the state of the art on our new dataset, we proceed in two ways.

First, we propose two baseline methods based on the state-of-the-art approaches (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). Note that our benchmark includes an order of magnitude more sequences compared to the recent articulated tracking datasets (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b) and the sequences in our benchmark are about five times longer, which makes it computationally expensive to run the graph partitioning on the full sequences as in the recent works (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). We modify these methods to make them applicable on our proposed dataset. The baselines and corresponding modifications are explained in Section 5.3.1.

Second, in order to broaden the scope of our evaluation we organized a *PoseTrack Challenge* in conjunction with ICCV'17 on our dataset by establishing an online

Submission	Pose model	Tracking model	Tracking granularity	mAP	MOTA
ProTracker Girdhar *et al.* (2017)	Mask R-CNN He *et al.* (2017)	Hungarian	pose-level	59.6	**51.8**
BUTD Jin *et al.* (2017)	PAF Cao *et al.* (2017)	graph partitioning	person-level and part-level	59.2	50.6
SOPT-PT Ma and Institute (2017)	PAF Cao *et al.* (2017)	Hungarian	pose-level	62.5	44.6
ML-LAB Zhu *et al.* (2017)	modified PAF Cao *et al.* (2017)	frame-to-frame assign.	pose-level	**70.3**	41.8
ICG Payer *et al.* (2017)	novel single-/multi-person CNN	frame-to-frame assign.	pose-level	51.2	32.0
ArtTrack-baseline	Faster-RCNN Huang *et al.* (2016) + + DeeperCut Insafutdinov *et al.* (2016a)	graph partitioning	pose-level	59.4	48.1
PoseTrack-baseline	PAF Cao *et al.* (2017)	graph partitioning	part-level	59.4	48.4

Table 5.1: Results of the top five pose tracking models on the PoseTrack 2017 dataset submitted to our evaluation server and of our baselines based on Insafutdinov *et al.* (2017) and Iqbal *et al.* (2017b). Note that mAP for some of the methods might be intentionally reduced to achieve higher MOTA (see discussion in text).

Submission	Pose model	Additional training data	mAP
ML-LAB Zhu *et al.* (2017)	modification of PAF Cao *et al.* (2017)	COCO	**70.3**
BUTDS Jin *et al.* (2017)	PAF Cao *et al.* (2017)	MPII-Pose + COCO	64.5
ProTracker Girdhar *et al.* (2017)	Mask R-CNN He *et al.* (2017)	COCO	64.1
SOPT-PT Ma and Institute (2017)	PAF Cao *et al.* (2017)	MPII-Pose + COCO	62.5
SSDHG	SSD Liu *et al.* (2016a) + + Hourglass Newell *et al.* (2016)	MPII-Pose + COCO	60.0
ArtTrack-baseline	DeeperCut	MPII-Pose + COCO	65.1
PoseTrack-baseline	PAF Cao *et al.* (2017)	COCO	59.4

Table 5.2: Results of the top five pose estimation models on the PoseTrack 2017 dataset submitted to the PoseTrack ICCV 2017 workshop and of our baselines. The methods are ordered according to mAP. Note that the mAP of ArtTrack and submission ProTracker Girdhar *et al.* (2017) is different from Tab. 5.1 because the evaluation in this table does not threshold detections by the score.

Model	Training Set	Head	Sho	Elb	Wri	Hip	Knee	Ank	mAP
ArtTrack-baseline	our dataset	73.1	65.8	55.6	47.2	52.6	50.1	44.1	55.5
ArtTrack-baseline	MPII	76.4	74.4	68.0	59.4	66.1	64.2	56.6	66.4
ArtTrack-baseline	MPII + our dataset	**78.7**	**76.2**	**70.4**	**62.3**	**68.1**	**66.7**	**58.4**	**68.7**

Table 5.3: Pose estimation performance (mAP) of our ArtTrack baseline for different training sets evaluated on the PoseTrack 2017 dataset.

Model	Head	Sho	Elb	Wri	Hip	Knee	Ank	Total	mAP
ArtTrack-baseline, $\tau = 0.1$	58.0	56.4	34.0	19.2	44.1	35.9	19.0	38.1	**68.6**
ArtTrack-baseline, $\tau = 0.5$	63.5	62.8	48.0	37.8	52.9	48.7	36.6	50.0	66.7
ArtTrack-baseline, $\tau = 0.8$	**66.2**	**64.2**	**53.2**	**43.7**	**53.0**	**51.6**	**41.7**	**53.4**	62.1

Table 5.4: Pose tracking performance (MOTA) of ArtTrack baseline for different part detection cut-off thresholds τ evaluated on the PoseTrack 2017 dataset.

evaluation server and inviting submissions from the research community. In the following we consider the top five methods submitted to the online evaluation server both for the pose estimation and pose tracking tasks. In Table 5.1 and 5.2 we list the best performing methods on each task sorted by MOTA and mAP, respectively. In the following we first describe our baselines based on the recent works (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b) and then summarize the main observations made in this evaluation.

5.3.1 Baseline Methods

We build the first baseline model following the graph partitioning formulation for articulated tracking proposed by Insafutdinov *et al.* (2017), but introduce two simplifications that follow Papandreou *et al.* (2017). First, we rely on a person detector to establish locations of people in the image and run pose estimation independently for each person detection. This allows us to deal with large variation in scale present in our dataset by cropping and rescaling images to canonical scale prior to pose estimation. In addition, this also allows us to group together the body-part estimates inferred for a given detection bounding box. As a second simplification we apply the model on the level of full body poses and not on the level of individual body parts as in our baselines (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). We use a publicly available Faster-RCNN (Ren *et al.*, 2015) detector from the TensorFlow Object Detection API (Huang *et al.*, 2016) for people detection. This detector has been trained on the "MS COCO" dataset and uses Inception-ResNet-V2 (Szegedy *et al.*, 2017) for image encoding. We adopt the DeeperCut CNN architecture from Chapter 3 as our pose estimation method. This architecture is based on the ResNet-101 converted to a fully convolutional network by removing the global pooling layer and utilizing atrous (or dilated) convolutions (Chen *et al.*, 2017a) to increase the resolution of the output scoremaps. Once all poses are extracted, we perform non-maximum suppression based on pose similarity criteria (Papandreou *et al.*, 2017) to filter out redundant person detections. We follow the cropping procedure of Papandreou *et al.* (2017) with the crop size 336x336px. Tracking is implemented as by Insafutdinov *et al.* (2017) by forming the graph that connects body-part hypotheses in adjacent frames and partitioning this graph into connected components using an approach by Levinkov *et al.* (2017). We use Euclidean distance between body joints to derive costs for graph edges. Such distance-based features were found to be effective by Insafutdinov *et al.* (2017) with additional features adding minimal improvements at the cost of substantially slower inference.

For the second baseline, we use the publicly available source code of Iqbal *et al.* (2017b) and replace the pose estimation model with the one by Cao *et al.* (2017). We empirically found that the pose estimation model by Cao *et al.* (2017) is better at handling large scale variations compared to DeeperCut (Insafutdinov *et al.*, 2016a) used in the original paper. We do not make any changes in the graph partitioning algorithm, but reduce the window size to 21 as compared to 31 used in the original model. We refer the readers to Iqbal *et al.* (2017b) for more details. The goal of constructing these strong baselines is to validate the results submitted to our evaluation server and to allow us to perform additional experiments presented in Section 5.4. In the remainder of this chapter, we refer to them as ArtTrack-baseline and PoseTrack-baseline respectively.

5.3.2 Main Observations

Two-stage design. The first observation is that all submissions follow a two-stage tracking-by-detection design. In the first stage, a combination of person detector and single-frame pose estimation method is used to estimate poses of people in each frame. The exact implementation of single-frame pose estimation method varies. Each of the top three articulated tracking methods builds on a different pose estimation approach (Mask-RCNN (He *et al.*, 2017), PAF (Cao *et al.*, 2017) and DeeperCut (Insafutdinov *et al.*, 2016a)). On the other hand, when evaluating methods according to pose estimation metric (see Table 5.2) three of the top four approaches build on PAF (Cao *et al.*, 2017). The performance still varies considerably among these PAF-based methods (70.3 for submission by ML-LAB Zhu *et al.* (2017) vs. 62.5 for submission by SOPT-PT Ma and Institute (2017)) indicating that large gains can be achieved within the PAF framework by introducing incremental improvements.

In the second stage the single-frame pose estimates are linked over time. For most of the methods the assignment is performed on the level of body poses, not individual parts. This is indicated in the "Tracking granularity" column in Table 5.1. Only submission by BUTD Jin *et al.* (2017) and our PoseTrack-baseline track people on the level of individual body parts. Hence, most methods establish correspondence/assembly of parts into body poses on the per-frame level. In practice, this is implemented by supplying a bounding box of a person and running pose estimation just for this box, then declaring maxima of the heatmaps as belonging together. This is suboptimal as multiple people overlap significantly, yet most approaches choose to ignore such cases (possibly for inference speed/efficiency reasons). The best performing approach by ProTracker Girdhar *et al.* (2017) relies on simple matching between frames based on Hungarian algorithm and matching cost based on intersection-over-union score between person bounding boxes. None of the methods is end-to-end in the sense that it is able to directly infer articulated people tracks from video. We observe that the pose tracking performance of the top five submitted methods saturates at around 50 MOTA, with the top four approaches showing rather similar MOTA results (51.8 for submission by ProTracker Girdhar *et al.* (2017) vs. 50.6 for submission by BUTD Jin *et al.* (2017) vs. 48.4 for PoseTrack-baseline vs. 48.1 for ArtTrack-baseline).

Training data. Most submissions found it necessary to combine our training set with datasets of static images such as COCO and MPII-Pose to obtain a joint training set with larger appearance variability. The most common procedure was to pre-train on external data and then fine-tune on our training set. Our training set is composed of 2437 people tracks with 61,178 annotated body poses and is complementary to COCO and MPII-Pose which include an order of magnitude more individual

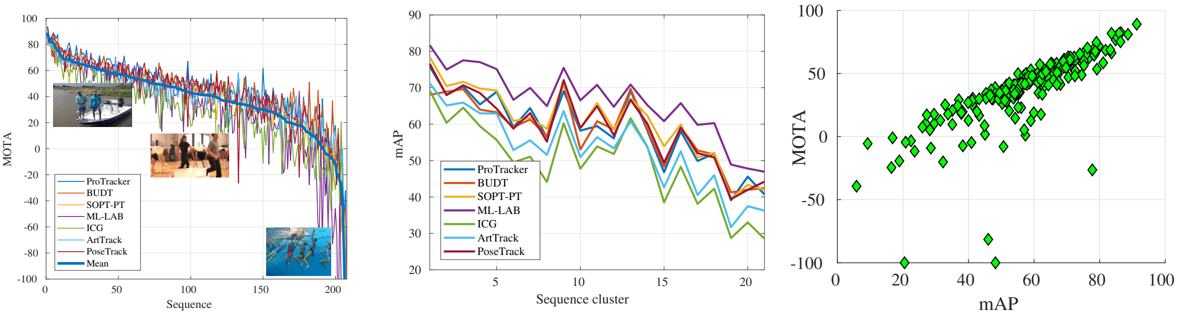


Figure 5.3: Sequences sorted by average MOTA (left). Pose estimation results sorted according to articulation complexity of the sequence (middle). Visualization of correlation between mAP and MOTA for each sequence (right). Note the outliers in right plot that correspond to sequences where pose estimation works well but tracking still fails.

people but do not provide motion information. We quantify the performance improvement due to training on additional data in Table 5.3 using our ArtTrack baseline. Extending the training data with the MPII-Pose dataset improves the performance considerably (55.5 vs. 68.7 mAP). The combination of our dataset and MPII-Pose still performs better than MPII-Pose alone (66.4 vs. 68.7) showing that datasets are indeed complementary.

None of the approaches in our evaluation employs any form of learning on the provided video sequences beyond simple cross-validation of a few hyperparameters. This can be in part due to relatively small size of our training set. One of the lessons learned from our work on this benchmark is that creating truly large annotated datasets of articulated pose sequences is a major challenge. We envision that future work will combine manually labeled data with other techniques such as transfer learning from other datasets such as the one proposed by Carreira and Zisserman (2017), inferring sequences of poses by propagating annotations from reliable keyframes (Charles *et al.*, 2016), and leveraging synthetic training data as in Varol *et al.* (2017).

Dataset difficulty. We composed our dataset by including videos around the keyframes from MPII Human Pose dataset that included several people and non-static scenes. The rationale was to create a dataset that would be non-trivial for tracking and require methods to correctly resolve effects such as person-person occlusions. In Figure 5.3 we visualize performance of the evaluated approaches on each of the test sequences. We observe that test sequences vary greatly with respect to difficulty both for pose estimation as well as for tracking. *E.g* , for the best performing submission by ProTracker Girdhar *et al.* (2017) the performance varies from nearly 80 MOTA to a score below zero[3]. Note that the approaches mostly agree with respect to the difficulty of the sequences. More difficult sequences are

[3] Note that MOTA metric can become negative for example when the number of false positives significantly exceeds the number of ground-truth targets.

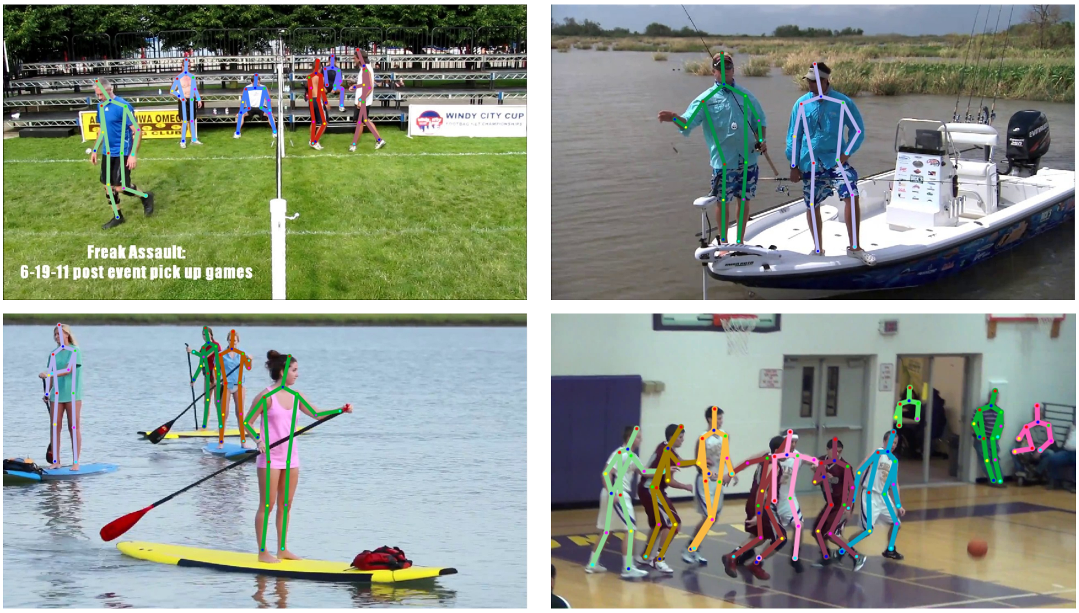


Figure 5.4: Selected frames from sample sequences with MOTA score above 75% with predictions of our ArtTrack-baseline overlaid in each frame. See text for further description.

likely to require methods that are beyond simple tracking component based on frame-to-frame assignment used in the currently best performing approaches. To encourage submissions that explicitly address challenges in the difficult portions of the dataset we have defined easy/moderate/hard splits of the data and report results for each of the splits as well as the full set.

Evaluation metrics. The MOTA evaluation metric has a deficiency in that it does not take the confidence score of the predicted tracks into account. As a result achieving good MOTA score requires tuning of the pose detector threshold so that only confident track and pose hypothesis are supplied for evaluation. This in general degrades pose estimation performance as measured by mAP (*c.f*. performance of submission by ProTracker Girdhar *et al.* (2017) in Table 5.1 and 5.2). We quantify this in Figure 5.4 for our ArtTrack baseline. Note that filtering the detections with score below $\tau = 0.8$ as compared to $\tau = 0.1$ improves MOTA from 38.1 to 53.4. One potential improvement to the evaluation metric would be to require that pose tracking methods assign confidence score to each predicted track as is common for pose estimation and object detection. This would allow one to compute a final score as an average of MOTA computed for a range of track scores. Current pose tracking methods typically do not provide such confidence scores. We believe that extending the evaluation protocol to include confidence scores is an important future direction.

Model	Head	Sho	Elb	Wri	Hip	Knee	Ank	Total	mAP
Submissions to the ICCV 2017 workshop									
ICG Payer *et al.* (2017)	55.4	47.9	25.8	17.8	24.2	22.4	18.5	32.0	51.2
ML-LAB Zhu *et al.* (2017)	57.3	52.0	37.0	31.1	41.2	37.7	28.3	41.8	70.3
SOPT-PT Ma and Institute (2017)	59.5	57.4	36.0	28.6	35.9	37.2	30.2	42.0	58.2
BUTD Jin *et al.* (2017)	64.6	63.3	49.6	41.7	49.8	44.7	33.4	50.6	59.2
ProTracker Girdhar *et al.* (2017)	58.2	61.0	53.3	44.6	50.2	49.1	43.1	51.8	59.6
Submissions to the ECCV 2018 workshop									
TML++ Hwang *et al.* (2019)	66.9	65.7	53.4	44.7	52.9	49.6	41.8	54.5	68.8
Simple Baselines Xiao *et al.* (2018)	67.1	68.4	52.2	48.9	56.1	56.6	48.8	57.6	73.9
PoseFlow Xiu *et al.* (2018)	52.0	57.4	52.8	46.6	51.0	51.2	45.3	51.0	63.0
JointFlow Doering *et al.* (2018)	65.8	66.0	51.7	41.7	53.5	47.3	39.2	53.1	63.6
STAF Raaj *et al.* (2019)	68.3	67.7	48.2	41.6	54.6	49.1	39.9	53.8	70.3
LightTrack Ning and Huang (2019)	60.7	65.6	59.9	55.4	56.6	57.1	49.4	58.0	66.7
HRNet Sun *et al.* (2019)	-	-	-	-	-	-	-	57.9	**74.9**
KeyTrack Snower *et al.* (2020)	67.0	69.3	60.0	57.7	58.1	59.7	53.3	61.2	74.0
DetTrack Wang *et al.* (2020)	**71.1**	**71.0**	**64.7**	**58.6**	**61.5**	**61.7**	**56.5**	**64.1**	74.1

Table 5.5: Multi-person pose estimation performance (mAP) of different methods on the PoseTrack 2017 test set.

Model	Assembly	(On/off)line	Pose Model	Tracking Model	MOTA
ICG Payer *et al.* (2017)	bottom-up	online	novel single-/multi-person CNN	frame-to-frame assign.	51.2
ML-LAB Zhu *et al.* (2017)	bottom-up	online	modified PAF Cao *et al.* (2017)	frame-to-frame assign.	70.3
SOPT-PT Ma and Institute (2017)	bottom-up	online	PAF Cao *et al.* (2017)	Hungarian	58.2
BUTD Jin *et al.* (2017)	bottom-up	offline	PAF Cao *et al.* (2017)	graph partitioning	59.2
ProTracker Girdhar *et al.* (2017)	top-down	online	Mask R-CNN He *et al.* (2017)	Hungarian	59.6
PoseFlow Xiu *et al.* (2018)	top-down	online	Faster R-CNN + Hourglass	frame-to-frame assign.	51.0
JointFlow Doering *et al.* (2018)	bottom-up	online	PAF Cao *et al.* (2017)	Hungarian (TAF)	53.1
STAF Raaj *et al.* (2019)	bottom-up	online	PAF Cao *et al.* (2017)	Hungarian (TAF)	53.8
Simple Baselines Xiao *et al.* (2018)	top-down	online	FPN-DCN Dai *et al.* (2017) + ResNet	Hungarian (Optical Flow)	57.6
LightTrack Ning and Huang (2019)	top-down	online	FPN-DCN Dai *et al.* (2017) + ResNet	frame-to-frame assign.	58.0
KeyTrack Snower *et al.* (2020)	top-down	online	HRNet Sun *et al.* (2019)	end-to-end + greedy	61.2
DetTrack Wang *et al.* (2020)	top-down	offline	HRNet Sun *et al.* (2019)	end-to-end + + tracklet merging	**64.1**

Table 5.6: Pose tracking performance (MOTA) of different methods on the PoseTrack 2017 test set. TAF denotes temporal affinity fields – a class of temporal scoring mechanisms proposed recently (Raaj *et al.*, 2019; Doering *et al.*, 2018).

	MOTA				
Model	Wrist	Ankles	Total	AP^T	AP
Submissions to the ECCV 2018 workshop					
MDPN Guo *et al.* (2018)	49.0	45.1	50.6	71.7	75.0
OpenSVAI Ning *et al.* (2018)			62.4	69.7	76.3
PT_CPN++ Yu *et al.* (2018)	61.2	56.7	64.0	-	80.9
TML++ Hwang *et al.* (2019)	56.4	52.4	65.7	74.6	-
STAF Raaj *et al.* (2019)	-	-	60.9	70.4	-
LightTrack Ning and Huang (2019)	-	-	64.6	72.4	77.2
KeyTrack Snower *et al.* (2020)	-	-	66.6	74.3	81.6
DetTrack Wang *et al.* (2020)	64.1	61.9	68.7	-	81.5

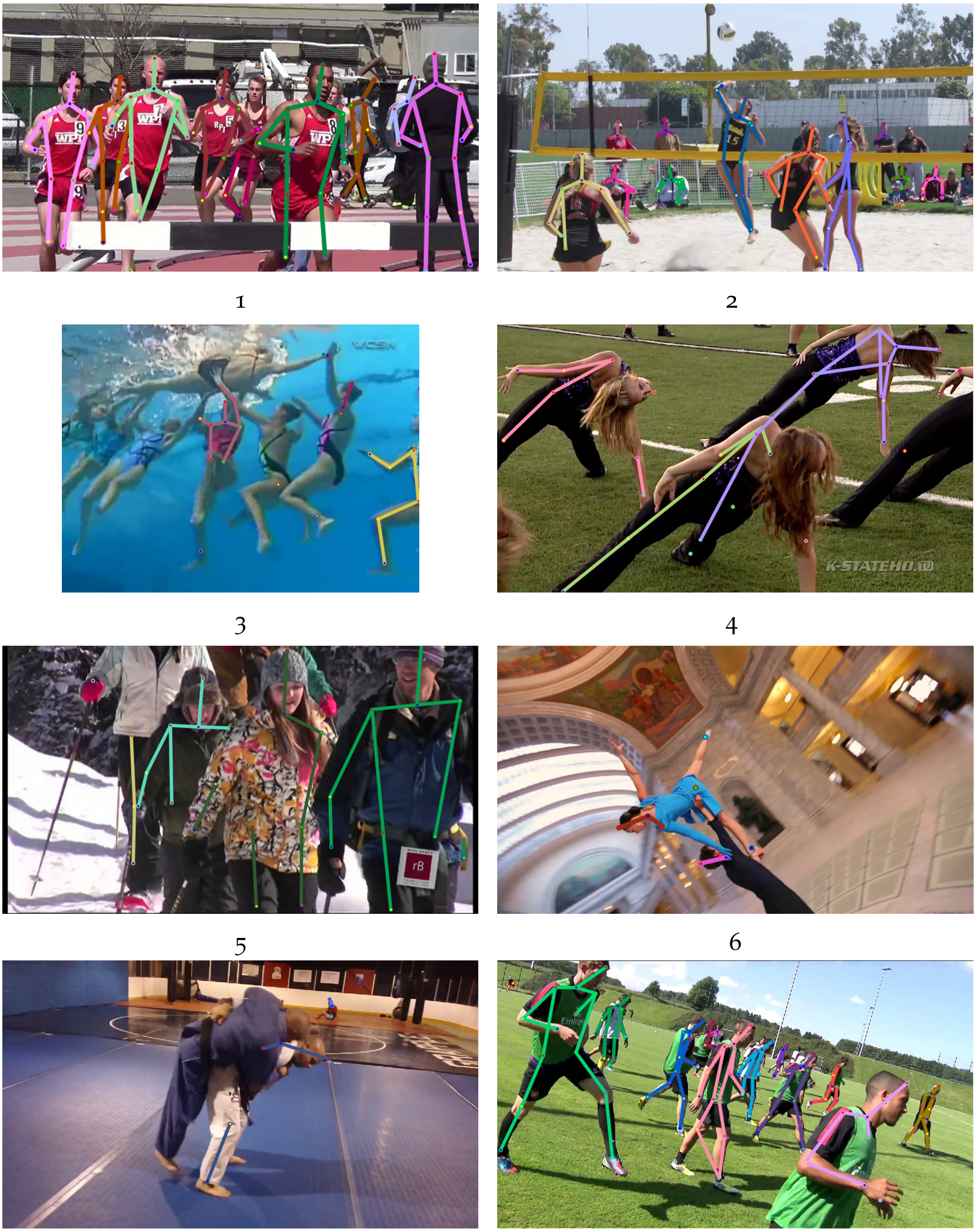
Table 5.7: Pose tracking performance (MOTA) of different methods on the PoseTrack 2018 validation set.

5.3.3 Recent State of the Art

Since the introduction of the dataset at ICCV 2017 workshop there has been a steady progress in the pose tracking accuracy as shown in Table 5.5. In about 2 years the pose tracking score (MOTA) improved by 15 percentage points. Upon release of the dataset state of the art methods did not learn temporal dynamics and instead relied on strong detectors and pose estimation algorithms that process frames individually and require post-hoc merging of poses into person tracks, either with greedy techniques (Girdhar *et al.*, 2018) or global optimization (Insafutdinov *et al.*, 2017; Iqbal *et al.*, 2017b). Linking poses over time was typically based on similarity measures such as bounding box IoU (Intersection-over-Union) or Object Keypoint Similarity (OKS) defined by the MS COCO keypoint detection benchmark (Lin *et al.*, 2014). Such approaches do not explicitely take advantage of the motion information available in video sequences.

Xiao *et al.* (2018) use optical flow to propagate joints to the current frame and compute OKS on the propagated coordinates, substantially improving accuracy over simple bounding box level similarity metrics. To account for potential occlusions the pose in the current frame is matched against multiple prior frames instead of only the immediate previous frame. Algorithms using optical flow can benefit from the advances in the optical flow literature by directly incorporating the latest algorithms. However, recent works (Neverova *et al.*, 2019; Bertasius *et al.*, 2019) show that learning to propagate poses directly can perform better than using optical flow. A different line of work (Doering *et al.*, 2018; Raaj *et al.*, 2019) investigates learning temporal fields that associate body joints between two consecutive frames.

Finally, the very recent approaches made important first steps towards end-to-end articulated tracking. Snower *et al.* (2020) propose to directly learn a similarity metric by training a classifier to predict whether two persons in different frames are the same instance. The classifier is implemented with a Transformer architecture (Vaswani *et al.*, 2017) operating directly on the keypoints output by a pose estimator, which is an easier task to be learned compared to using raw pixels. The current state of the art on the PoseTrack dataset is held by a method of Wang *et al.* (2020) (see Tables 5.5 and 5.7). To the best of our knowledge it is the first method that performs pose tracking in an end-to-end fashion. It operates in top-down manner and runs a person detector for every frame. For every detected box a tublet is cut out, centered on the corresponding frame, and extending both backwards and forwards in time. Finally, they train a network that predicts a pose tracklet given the tubelet as input. As this is done for every detection in every frame, the generated set of tracklets is overcomplete and the authors design a procedure for merging tracklets into final person tracks. This work significantly outperforms all prior methods by a substantial margin and demonstrates that end-to-end learning is essential for modeling temporal dynamics of human motion. This work also opens an avenue for research on more efficient end-to-end tracking, as their approach performs a lot of redundant computation.



1

2

3

4

5

6

7

8

Figure 5.5: Selected frames from sample sequences with negative average MOTA score. The predictions of our ArtTrack-baseline are overlaid in each frame. Challenges for current methods in such sequences include crowds (images 3 and 8), extreme proximity of people to each other (7), rare poses (4 and 6) and strong camera motions (3, 5, 6, and 8).

5.4 DATASET ANALYSIS

In order to better understand successes and failures of the current body pose tracking approaches, we analyze their performance across the range of sequences in the test set. To that end, for each sequence we compute an average over MOTA scores obtained by each of the seven evaluated methods. Such average score serves us as an estimate for the difficulty of the sequence for the current computer vision approaches. We then rank the sequences by the average MOTA. The resulting ranking is shown in Figure 5.3 (left) along with the original MOTA scores of each of the approaches. First, we observe that all methods perform similarly well on easy sequences. Figure 5.4 shows a few easy sequences with an average MOTA above 75%. Visual analysis reveals that easy sequences typically contain significantly separated individuals in upright standing poses with minimal changes of body articulation over time and no camera motion. Tracking accuracy drops with the increased complexity of video sequences. Figure 5.5 shows a few hard sequences with average MOTA accuracy below 0. These sequences typically include strongly overlapping people, and fast motions of people and camera.

We further analyze how tracking and pose estimation accuracy are affected by pose complexity. As a measure for the pose complexity of a sequence we employ an average deviation of each pose in a sequence from the mean pose. The computed complexity score is used to sort video sequences from low to high pose complexity and average mAP is reported for each sequence. The result of this evaluation is shown in Figure 5.3 (middle). For visualization purposes, we partition the sorted video sequences into bins of size 10 based on pose complexity score and report average mAP for each bin. We observe that both body pose estimation and tracking performance significantly decrease with the increased pose complexity. Figure 5.3 (right) shows a plot that highlights correlation between mAP and MOTA of the same sequence. We use the mean performance of all methods in this visualization. Note that in most cases more accurate pose estimation reflected by higher mAP indeed corresponds to higher MOTA. However, it is instructive to look at sequences where poses are estimated accurately (mAP is high), yet tracking results are particularly poor (MOTA near zero). One of such sequences is shown in Figure 5.5 (8). This sequence features a large number of people and fast camera movement that is likely confusing simple frame-to-frame association tracking of the evaluated approaches. Please see supplemental material for additional examples and analyses of challenging sequences.

5.5 CONCLUSION

In this chapter we proposed a new benchmark for human pose estimation and articulated tracking that is significantly larger and more diverse in terms of data variability and complexity compared to existing pose tracking benchmarks. Our benchmark enables objective comparison of different approaches for articulated people tracking in realistic scenes. We have set up an online evaluation server that permits evaluation on a held-out test set, and have measures in place to limit overfitting on the dataset. Finally, we conducted a rigorous survey of the state of the art. Due to the scale and complexity of the benchmark, most existing methods build on combinations of proven components: people detection, single-person pose estimation, and tracking based on simple association between neighboring frames. Our analysis shows that current methods perform well on easy sequences with well separated upright people, but are severely challenged in the presence of fast camera motions and complex articulations. Addressing these challenges remains an important direction for the future work.

Part II

Reconstructing 3D Human Pose and Object Shape

6 EGOCENTRIC MARKER-LESS MOTION CAPTURE

Contents

Marker-based and marker-less optical skeletal motion-capture methods use an *outside-in* arrangement of cameras placed around a scene, with viewpoints converging on the center. They often create discomfort with marker suits, and their recording volume is severely restricted and often constrained to indoor scenes with controlled backgrounds. Alternative suit-based systems use several inertial measurement units or an exoskeleton to capture motion with an *inside-in* setup, i.e. without external sensors. This makes capture independent of a confined volume, but requires substantial, often constraining, and hard to set up body instrumentation. This chapter presents a new method for real-time, marker-less, and egocentric motion capture: estimating the full-body skeleton pose from a lightweight stereo pair of fisheye cameras attached to a helmet or virtual reality headset – an *optical inside-in* method, so to speak. This allows full-body motion capture in general indoor and outdoor scenes, including crowded scenes with many people nearby, which enables reconstruction in larger-scale activities. The approach combines the strength of a new generative pose estimation framework for fisheye views with a CNN-based body-part detector trained on a large new dataset. It is particularly useful in virtual reality to freely roam and interact, while seeing the fully motion-captured virtual body.

6.1 INTRODUCTION

Traditional optical skeletal motion-capture methods – both marker-based and marker-less – use several cameras typically placed around a scene in an *outside-in* arrangement, with camera views approximately converging in the center of a confined recording volume. This greatly constrains the spatial extent of motions that can be recorded; simply enlarging the recording volume by using more cameras, for instance to capture an athlete, is not scalable. Outside-in arrangements also constrain the type of scene that can be recorded, even if it fits into a confined space. If a recording location is too small, cameras can often not be placed sufficiently far away. In other cases, a scene may be cluttered with objects or furniture, or other dynamic scene elements, such as people in close interaction, may obstruct a motion-captured person in the scene or create unwanted dynamics in the background. In such cases, even state-of-the-art outside-in marker-less optical methods that succeed with just a few cameras, and are designed for less controlled and outdoor scenes (Elhayek *et al.*, 2015), quickly fail. Scenes with dense social interaction were previously captured with outside-in camera arrays of a few hundred sensors (Joo *et al.*, 2015), a very complex and difficult to scale setup.

These strong constraints on recording volume and scene density prevent the use of optical motion capture in the majority of real-world scenes. This problem can partly be bypassed with *inside-in* motion-capture methods that use body-worn sensors exclusively (Menache, 2010), such as the Xsens MVN inertial measurement unit suit. However, the special suit and cabling are obstructive and require tedious calibration. Shiratori *et al.* (2011) propose to wear 16 cameras placed on body parts facing *inside-out*, and capture the skeletal motion through structure-from-motion relative to the environment. This clever solution requires instrumentation, calibration and a static background, but allows free roaming. This design was inspirational for our egocentric approach.

We propose EgoCap: an egocentric motion-capture approach that estimates full-body pose from a pair of optical cameras carried by lightweight headgear (see Figure 6.1). The body-worn cameras are oriented such that their field of view covers the user's body entirely, forming an arrangement that is independent of external sensors – an *optical inside-in* method, if you will. We show that our optical full-body approach overcomes many limitations of existing outside-in, inside-out and IMU-based inside-in methods. It reduces the setup effort, enables free roaming, and minimizes body instrumentation. EgoCap decouples the estimation of local body pose with respect to the headgear cameras and global headgear position, which we

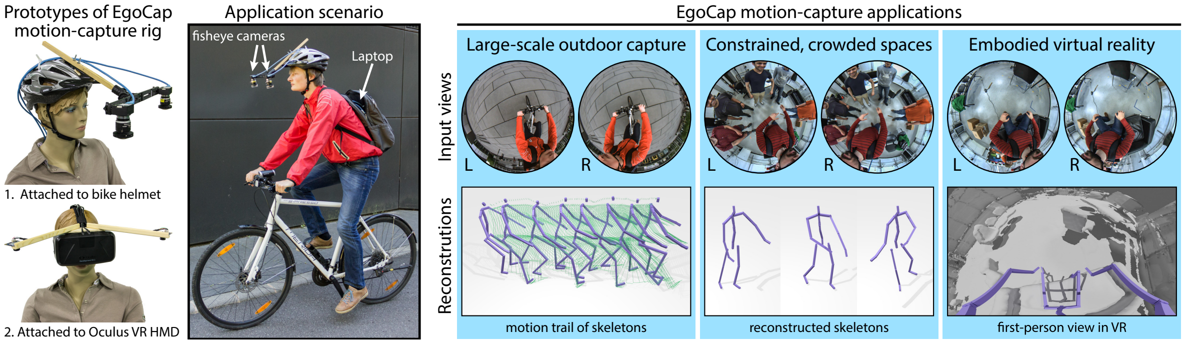

Prototypes of EgoCap motion-capture rig
1. Attached to bike helmet
2. Attached to Oculus VR HMD
Application scenario
fisheye cameras
Laptop
EgoCap motion-capture applications
Large-scale outdoor capture
Constrained, crowded spaces
Embodied virtual reality
Input views
Reconstrutions
L
R
L
R
L
R
motion trail of skeletons
reconstructed skeletons
first-person view in VR

Figure 6.1: We propose a marker-less optical motion-capture approach that only uses two head-mounted fisheye cameras (see rigs on the left). Our approach enables three new application scenarios: (1) capturing human motions in outdoor environments of virtually unlimited size, (2) capturing motions in space-constrained environments, e.g. during social interactions, and (3) rendering the reconstruction of one's real body in virtual reality for embodied immersion.

infer by inside-out structure-from-motion on the scene background.

Our first contribution is a new egocentric inside-in sensor rig with only two head-mounted, downward-facing commodity video cameras with fisheye lenses (see Figure 6.1). While head-mounted cameras might pose a problem with respect to social acceptance and ergonomics in some scenarios, performances have not been hindered during our recordings and VR tests. The rig can be attached to a helmet or a head-mounted VR display, and, hence, requires less instrumentation and calibration than other body-worn systems. The stereo fisheye optics keep the whole body in view in all poses, despite the cameras' proximity to the body. We prefer conventional video cameras over IR-based RGB-D cameras, which were for example used for egocentric hand tracking (Sridhar *et al.*, 2015), as video cameras work indoors and outdoors, have lower energy consumption and are easily fitted with the required fisheye optics.

Our second contribution is a new marker-less motion capture algorithm tailored to the strongly distorted egocentric fisheye views. It combines a generative model-based skeletal pose estimation approach (6.4) with evidence from a trained CNN-based body part detector (6.4.3). The approach features an analytically differentiable objective energy that can be minimized efficiently, is designed to work with unsegmented frames and general backgrounds, succeeds even on poses exhibiting notable self-occlusions (e.g. when walking), as the part detector predicts occluded parts, and enables recovery from tracking errors after severe occlusions.

Our third contribution is a new approach for automatically creating body part detection training datasets. We record test subjects in front of green screen with an existing outside-in marker-less motion capture system to get ground-truth skeletal poses, which are reprojected into the simultaneously recorded head-mounted fisheye views to get 2D body part annotations. We augment the training images by replacing the green screen with random background images, and vary the appearance in terms of color and shading by intrinsic recoloring (Meka *et al.*, 2016). With this technique, we annotate a total of 100,000 egocentric images of eight people in different clothing (6.4.3.1), with 75,000 images from six people used for training. We publish the dataset for research purposes (EgoCap, 2016).

We designed and extensively tested two system prototypes featuring (1) cameras fitted to a bike helmet, and (2) small cameras attached to an Oculus Rift headset. We show reliable egocentric motion capture, both off-line and in real time. The egocentric tracking meets the accuracy of outside-in approaches using 2–3 cameras; additional advances are necessary to match the accuracy of many-camera systems. In our egocentric setup, reconstructing the lower body is more challenging due to its larger distance and frequent occlusions, and is less accurate compared to the upper body in our experiments. Nevertheless, we succeed in scenes that are challenging for outside-in approaches, such as close interaction with many people, as well outdoor and indoor scenes in cluttered environments with frequent occlusions, for example when working in a kitchen or at a desk. We also show successful capturing in large volumes, for example of the skeletal motion of a cyclist. The lightweight Oculus Rift gear is designed for egocentric motion capture for virtual reality, where the user can move in the real world to roam and interact in a virtual environment seen through a head-mounted display, while perceiving increased immersion thanks to the rendering of the motion-captured body, which is not obtained with current HMD head pose tracking.

6.2 RELATED WORK

Suit-based Motion Capture Marker-based optical systems use a suit with passive retro-reflective spheres (e.g. Vicon) or active LEDs (e.g. PhaseSpace). Skeleton motion is reconstructed from observed marker positions in multiple cameras (usually 10 or more) in an outside-in arrangement, producing highly accurate sparse motion data, even of soft tissue (Park and Hodgins, 2008; Loper *et al.*, 2014), but the external cameras severely restrict the recording volume. For character animation purposes, where motions are restricted, use of motion sub-spaces can reduce requirements to six markers and two cameras (Chai and Hodgins, 2005), or a single foot pressure-sensor pad (Yin and Pai, 2003), which greatly improves usability. For hand tracking, a color glove and one camera (Wang and Popović, 2009) is highly practical. Inertial measurement units (IMUs) fitted to a suit (e.g. Xsens MVN) allow free roaming and high reliability in cluttered scenes by inside-in motion capture, i.e. without requiring external sensors (Tautges *et al.*, 2011). Combinations with ultrasonic distance sensors (Vlasic *et al.*, 2007), video input (Pons-Moll *et al.*, 2010, 2011), and pressure plates Ha *et al.* (2011) suppress the drift inherent to IMU measurements and reduce the number of required IMUs. Besides drift, the instrumentation with IMU sensors is the largest drawback, causing long setup times and intrusion. Exoskeleton suits (e.g. METAmotion Gypsy) avoid drift, but require more cumbersome instrumentation. Turning the standard outside-in capturing approach on its head, Shiratori *et al.* (2011)

attach 16 cameras to body segments in an inside-out configuration, and estimate skeletal motion from the position and orientation of each camera as computed with structure-from-motion. This clever solution – which was inspirational for our egocentric approach – allows free roaming although it requires instrumentation and a static background.

Marker-less Motion Capture Recent years have seen great advances in marker-less optical motion-capture algorithms that track full-body skeletal motions, reaching and outperforming the reconstruction quality of suit- and marker-based approaches (Bregler and Malik, 1998; Theobalt *et al.*, 2010; Moeslund *et al.*, 2011; Holte *et al.*, 2012). Marker-less approaches also typically use an outside-in camera setup, and were traditionally limited to controlled studio environments, or scenes with static, easy-to-segment background, using 8 or more cameras (e.g. Urtasun *et al.*, 2006; Gall *et al.*, 2010; Sigal *et al.*, 2010, 2012; Stoll *et al.*, 2011). Recent work is moving towards less controlled environments and outdoor scenes, also using fewer cameras (Amin *et al.*, 2009; Burenius *et al.*, 2013; Elhayek *et al.*, 2015; Rhodin *et al.*, 2015), but still in an outside-in configuration. These approaches are well-suited for static studio setups, but share the limitation of constrained recording volumes, and reach their limits in dense, crowded scenes. Joo *et al.* (2015) use a camera dome with 480 outside-in cameras for motion capture of closely interacting people, but domes do not scale to larger natural scenes.

Motion Capture with Depth Sensors 3D pose estimation is highly accurate and reliable when using multiple RGB-D cameras Zhang *et al.* (2014), and even feasible from a single RGB-D camera in real time (e.g. Shotton *et al.*, 2011b; Baak *et al.*, 2011; Wei *et al.*, 2012). However, many active IR-based depth cameras are unsuitable for outdoor capture, have high energy consumption, and equipping them with fisheye optics needed for our camera placement is hard.

Egocentric Motion Capture In the past, egocentric inside-in camera placements were used for tracking or model learning of certain parts of the body, for example of the face with a helmet-mounted camera or rig (Jones *et al.*, 2011; Wang *et al.*, 2016), of fingers from a wrist-worn camera (Kim *et al.*, 2012), or of eyes and eye gaze from cameras in a head-mounted rig (Sugano and Bulling, 2015). Rogez *et al.* (2014) and Sridhar *et al.* (2015) track articulated hand motion from body- or chest-worn RGB-D cameras. Using a body-worn depth camera, Yonemoto *et al.* (2015) extrapolate arm and torso poses from arm-only RGB-D footage. Jiang and Grauman (2016) attempted full-body pose estimation from a chest-worn camera view by analyzing the scene, but without observing the user directly and at very restricted accuracy. Articulated full-body motion capture with a lightweight head-mounted camera pair was not yet attempted.

First-person Vision A complementary research branch analyses the environment from first-person, i.e. body-worn outward-facing cameras, for activity recognition

(Fathi *et al.*, 2011; Kitani *et al.*, 2011; Ohnishi *et al.*, 2016; Ma *et al.*, 2016), for learning engagement and saliency patterns of users when interacting with the real world (e.g. Park *et al.*, 2012; Su and Grauman, 2016), and for understanding the utility of surrounding objects (Rhinehart and Kitani, 2016). Articulated full-body tracking, or even only arm tracking, is not their goal, but synergies of both fields appear promising.

2D and 3D Pose Detection Traditionally, 2D human pose estimation from monocular images is a two-stage process where coherent body pose is inferred from local image evidence Yang and Ramanan (2012); Johnson and Everingham (2011). Convolutional networks brought a major leap in performance Chen and Yuille (2014); Jain *et al.* (2014a,b); Tompson *et al.* (2014); Toshev and Szegedy (2014) and recent models demonstrated that end-to-end prediction is possible due to the large receptive fields capturing the complete pose context (Pishchulin *et al.*, 2016). Pfister *et al.* (2015) and Wei *et al.* (2016) allow for increased depth and learning of spatial dependencies between body parts by layering multiple CNNs. We adopt the body part detectors introduced in the Chapter 3, which builds on the recent success of residual networks (He *et al.*, 2016; Newell *et al.*, 2016), which further facilitate an increase in network depth. Recently, direct 3D pose estimation has emerged by lifting 2D poses to 3D (Yasin *et al.*, 2016), using mid-level posebit descriptors (Pons-Moll *et al.*, 2014), and motion compensation in videos (Tekin *et al.*, 2016), but estimates are still coarse. Existing detection methods use simplified body models with few body parts to reduce the enormous cost of creating sufficiently large, annotated training datasets, do not generalize to new camera geometry and viewpoints, such as egocentric views, and results usually exhibit jitter over time.

6.3 EGOCENTRIC CAMERA DESIGN

We designed a mobile egocentric camera setup to enable human motion capture within a virtually unlimited recording volume. We attach two fisheye cameras rigidly to a helmet or VR headset, such that their field of view captures the user's full body, see 6.2. The wide field of view allows to observe interactions in front and beside the user, irrespective of their global motion and head orientation, and without requiring additional sensors or suits. The stereo setup ensures that most actions are observed by at least one camera, despite substantial self-occlusions of arms, torso and legs in such an egocentric setup. A baseline of 30–40 cm proved to be best in our experiments. The impact of the headgear on the user's motion is limited as it is lightweight: our prototype camera rig for VR headsets (see 6.1, bottom left) only adds about 65 grams of weight.

6.4 EGOCENTRIC FULL-BODY MOTION CAPTURE

Our egocentric setup separates human motion capture into two subproblems: (1) local skeleton pose estimation with respect to the camera rig, and (2) global rig pose estimation relative to the environment. Global pose is estimated with existing structure-from-motion techniques, see Section 6.6.3. We formulate skeletal pose estimation as an analysis-by-synthesis-style optimization problem in the pose parameters $\mathbf{p}^t$, that maximizes the alignment of a projected 3D human body model (Section 6.4.1) with the human in the left $\mathcal{I}^t_{\text{left}}$ and the right $\mathcal{I}^t_{\text{right}}$ stereo fisheye views, at each video time step t. We use a hybrid alignment energy combining evidence from a generative image-formation model, as well as from a discriminative detection approach. Our generative ray-casting-based image formation model is inspired by light transport in volumetric translucent media, and enables us to formulate a color-based alignment term in $\mathbf{p}^t$ that is analytically differentiable and features an analytically differentiable formulation of 3D visibility (Section 6.4.2). This model facilitates generative pose estimation with only two cameras, and we adapt it to the strongly distorted fisheye views. Our energy also employs constraints from one-shot joint-location predictions in the form of $E_{\text{detection}}$. These predictions are found with a new CNN-based 2D joint detector for head-mounted fisheye views, which is learned from a large corpus of annotated training data, and which generalizes to different users and cluttered scenes (Section 6.4.3). The combined energy that we optimize takes the following form:

$$E(\mathbf{p}^t) = E_{\text{color}}(\mathbf{p}^t) + E_{\text{detection}}(\mathbf{p}^t) + E_{\text{pose}}(\mathbf{p}^t) + E_{\text{smooth}}(\mathbf{p}^t). \tag{6.1}$$

Here, $E_{\text{pose}}(\mathbf{p}^t)$ is a regularizer that penalizes violations of anatomical joint-angle limits as well as poses deviating strongly from the rest pose ($\mathbf{p} = \mathbf{0}$):

$$\begin{aligned} E_{\text{pose}}(\mathbf{p}^t) = \lambda_{\text{limit}} \cdot \Big(&\max(0, \mathbf{p}^t - \mathbf{l}_{\text{upper}})^2 + \max(0, \mathbf{l}_{\text{lower}} - \mathbf{p}^t)^2\Big) \\ &+ \lambda_{\text{pose}} \cdot \text{huber}(\mathbf{p}^t), \end{aligned} \tag{6.2}$$

where $\mathbf{l}_{\text{lower}}$ and $\mathbf{l}_{\text{upper}}$ are lower and upper joint-angle limits, and $\text{huber}(x) = \sqrt{1+x^2} - 1$ is the Pseudo-Huber loss function. $E_{\text{smooth}}(\mathbf{p}^t)$ is a temporal smoothness term:

$$E_{\text{smooth}}(\mathbf{p}^t) = \lambda_{\text{smooth}} \cdot \text{huber}(\mathbf{p}^{t-1} + \zeta(\mathbf{p}^{t-1} - \mathbf{p}^{t-2}) - \mathbf{p}^t), \tag{6.3}$$

where $\zeta = 0.25$ is a damping factor. The total energy in Equation 6.1 is optimized for every frame, as described in Section 6.4.4. In the following, we describe the generative and discriminative terms in more detail, while omitting the temporal dependency t in the notation for better readability.

We use weights $\lambda_{\text{pose}} = 10^{-4}$, $\lambda_{\text{limit}} = 0.1$ and $\lambda_{\text{smooth}} = 0.1$.

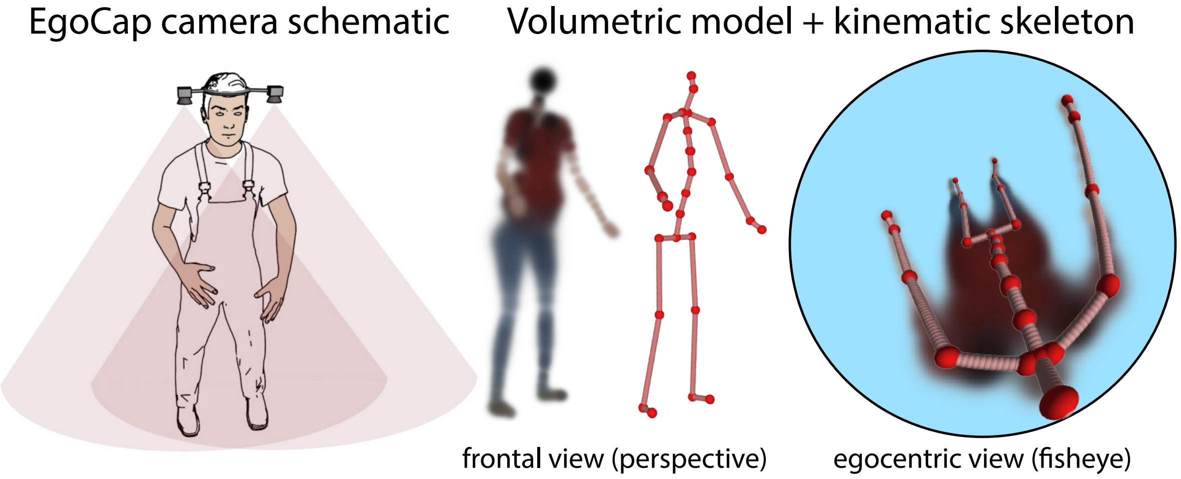

EgoCap camera schematic
Volumetric model + kinematic skeleton
frontal view (perspective)
egocentric view (fisheye)

Figure 6.2: Schematic of EgoCap, our egocentric motion-capture rig (left), visualization of the corresponding volumetric body model and kinematic skeleton (center), and the egocentric view of both in our head-mounted fisheye cameras (right).

6.4.1 Body Model

We model the 3D body shape and pose of humans in 3D using the approach proposed by **?**, which represents the body volumetrically as a set of $N_q = 91$ isotropic Gaussian density functions distributed in 3D space. Each Gaussian G_q is parametrized by its standard deviation σ_q, location $\boldsymbol{\mu}_q$ in 3D space, density c_q and color $\mathbf{a}_q$, which define the Gaussian shape parameters. The combined density field of the Gaussians, $\sum_q c_q G_q$, smoothly describes the volumetric occupancy of the human in 3D space, see Figure 6.2. Each Gaussian is rigidly attached to one of the bones of an articulated skeleton with 17 joints, whose pose is parameterized by 37 twist pose parameters (Murray *et al.*, 1994).

Shape and skeleton bone lengths need to be personalized to the tracked user prior to capturing. Commercial systems often use a dedicated initialization sequence at the start. Research papers on marker-less motion capture often treat initialization as a separate problem, and initialize models manually, which we could also do. However, we propose a much more automated initialization procedure to reduce setup time and effort. To this end, we adapt the approach of Rhodin *et al.* (2016), who personalize a 3D parametric human shape model of Gaussian density and skeleton dimensions by fitting it to multi-view images using a volumetric contour alignment energy. We adapt this to our stereo fisheye setting. In our egocentric setup 3–4 different user poses, showing the bending of knees, elbows and wrists without any occlusion, were sufficient for automatic shape and skeleton personalization, and only the automatically inferred Gaussian colors are manually corrected on body parts viewed at acute angles.

6.4.2 Egocentric Volumetric Ray-Casting Model

For color-based model-to-image similarity, we use the ray-casting image formation model of the previously described volumetric body model (Rhodin *et al.*, 2015). We first describe image formation assuming a standard pinhole model, as in Rhodin *et al.*, and then describe how we modify it for fisheye views. A ray is cast from the camera center c in direction $\mathbf{n}$ of an image pixel. The visibility of a particular 3D Gaussian G_q along the ray $(c + s\mathbf{n})$ is computed via

$$\mathcal{V}_q(c,\mathbf{n},\mathbf{p}) = \int_0^\infty \exp\left(-\int_0^s \sum_i G_i(c+t\mathbf{n})\,\mathrm{d}t\right) G_q(c+s\mathbf{n})\,\mathrm{d}s. \tag{6.4}$$

This formulation of visibility and color of a 3D Gaussian from the camera view is based on a model of light transport in heterogeneous translucent media Cerezo *et al.* (2005). $\mathcal{V}_q$ is the fraction of light along the ray that is absorbed by Gaussian G_q. We use this image-formation model in an energy term that computes the agreement of model and observation by summing the visibility-weighted color dissimilarity $d(\cdot,\cdot)$, which we explain later, between image pixel color $\mathcal{I}(u,v)$ and the Gaussian's color $\mathbf{a}_q$:

$$E_{\text{color}}(\mathbf{p},\mathcal{I}) = \sum_{(u,v)} \sum_q d(\mathcal{I}(u,v),\mathbf{a}_q)\mathcal{V}_q(c,\mathbf{n}(u,v),\mathbf{p}). \tag{6.5}$$

Note that this formulation has several key advantages over previous generative models for image-based pose estimation. It enables analytic derivatives of the pose energy, including a smooth analytically differentiable visibility model everywhere in pose space. This makes it perform well with only a few camera views. Previous methods often used fitting energies that are non-smooth or even lacking a closed-form formulation, requiring approximate recomputation of visibility (e.g. depth testing) inside an iterative optimization loop. Rhodin *et al.*'s formulation forms a good starting point for our egocentric tracking setting, as non-stationary backgrounds and occlusions are handled well. However, it applies only to static cameras, does not support the distortion of fisheye lenses, and it does not run in real time.

Color Dissimilarity For measuring the dissimilarity $d(\mathbf{m},\mathbf{i})$ of model color $\mathbf{m}$ and image pixel color $\mathbf{i}$ in Equation 6.5, we use the HSV color space (with all dimensions normalized to unit range) and combine three dissimilarity components:

1. For saturated colors, the color dissimilarity d_s is computed using the squared (minimum angular) hue distance. Using the hue channel alone gains invariance to illumination changes.

2. For dark colors, the color dissimilarity d_d is computed as twice the squared value difference, i.e. $d_d(\mathbf{m},\mathbf{i}) = 2(m_v - i_v)^2$. Hue and saturation are ignored as they are unreliable for dark colors.

3. For gray colors, the distance d_g is computed as the sum of absolute value and saturation difference, i.e. $d_g(\mathbf{m},\mathbf{i}) = |m_v - i_v| + |m_s - i_s|$. Hue is unreliable and thus ignored.

We weight these three dissimilarity components by $w_s = \sqrt{m_s}/Z$, $w_d = \max(0, 0.5 - m_v)/Z$ and $w_g = \max(0, 0.5 - m_s)/Z$ respectively, where Z normalizes the sum of these weights to unity. The total dissimilarity is computed by $d(\mathbf{m},\mathbf{i}) = \phi(w_s d_s + w_d d_d + w_g w_g)$ where $\phi(x) = 1 - (1-x)^4(8x+2)$ is a smooth step function. We employ a two-sided energy, i.e. E_{color} can be negative: For dissimilar colors, $d \approx 1$ and approaches -1 for similar colors.

6.4.2.1 *Egocentric Ray-Casting Model*

In our egocentric camera rig, the cameras move rigidly with the user's head. In contrast to commonly used skeleton configurations, where the hip is taken as the root joint, our skeleton hierarchy is rooted at the head. Like a puppet, the lower body parts are then relative to the head motion, see Figure 6.2. This formulation factors out the user's global motion, which can be estimated independently, see Section 6.6.3, and reduces the dimensionality of the pose estimation by 6 degrees of freedom. By attaching the cameras to the skeleton root, the movable cameras are reduced to a static camera formulation such that Equation 6.4 applies without modification.

Simply undistorting the fisheye images before optimization is impractical as resolution at the image center reduces and pinhole cameras cannot capture fields of view approaching 180 degrees – their image planes would need to be infinitely large. To apply the ray-casting formulation described in the previous section to our egocentric motion-capture rig, with its 180° field of view, we replace the original pinhole camera model with the omnidirectional camera model of Scaramuzza *et al.* (2006). The ray direction $\mathbf{n}(u,v)$ of a pixel (u,v) is then given by $\mathbf{n}(u,v) = [u, v, f(\rho)]^\top$, where f is a polynomial of the distance ρ of (u,v) to the estimated image center. We combine the energy terms for the two cameras (Equation 6.5) in our egocentric camera rig using

$$E_{\text{color}}(\mathbf{p}) = E_{\text{color}}(\mathbf{p}, \mathcal{I}_{\text{left}}) + E_{\text{color}}(\mathbf{p}, \mathcal{I}_{\text{right}}). \tag{6.6}$$

These extensions also generalize the contour model of Rhodin *et al.* (2016) to enable egocentric body model initialization.

6.4.3 Egocentric Body-Part Detection

We combine the generative model-based alignment from the previous section with evidence from the discriminative joint-location detector introduced in the Chapter 3, trained on annotated egocentric fisheye images. The discriminative component dramatically improves the quality and stability of reconstructed poses, provides efficient recovery from tracking failures, and enables plausible tracking even under notable self-occlusions. To apply our body-part detector, which has shown state-of-the-art results on human pose estimation from outside-in RGB images, to the

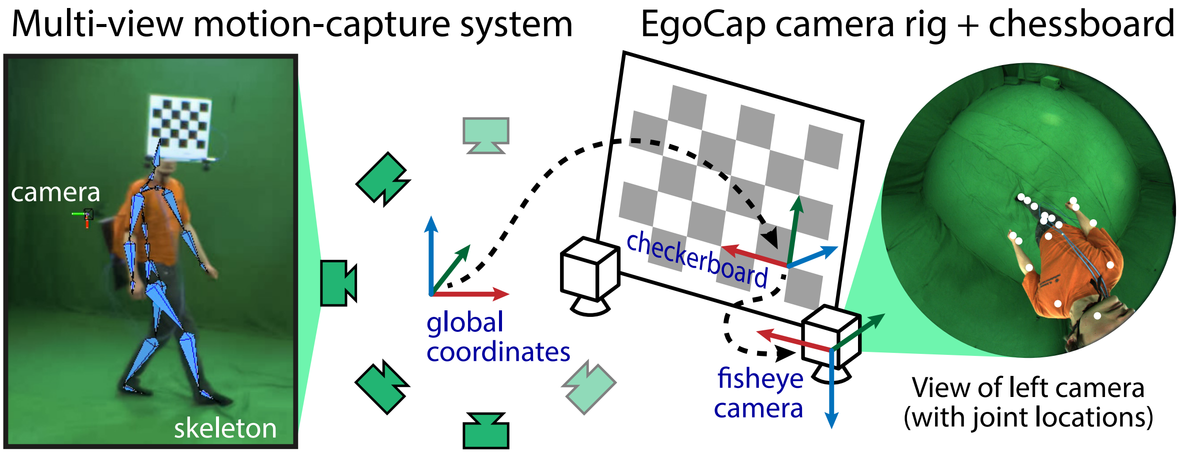

Multi-view motion-capture system
EgoCap camera rig + chessboard
camera
skeleton
global coordinates
checkerboard
fisheye camera
View of left camera (with joint locations)

Figure 6.3: For database annotation, the skeleton estimated from the multi-view motion capture system (left), is converted from global coordinates (center) into each fisheye camera's coordinate system (right) via the checkerboard.

top-down perspective and fisheye distortion of our novel egocentric camera setup, the largest burden is to gather and annotate a training dataset that is sufficiently large and varied, containing tens of thousands of images. As our camera rig is novel, there are no existing public datasets, and we therefore designed a method to automatically annotate real fisheye images by outside-in motion capture and to augment appearance with the help of intrinsic image decomposition.

6.4.3.1 *Dataset Creation*

We propose a novel approach for semi-automatically creating large, realistic training datasets for body-part detection that comprise tens of thousands of camera images annotated with the joint locations of a kinematic skeleton and other body parts such as the hands and feet. To avoid the tedious and error-prone manual annotation of locations in thousands of images, as in previous work, we use a state-of-the-art marker-less motion capture system (Captury Studio of The Captury) to estimate the skeleton motion in 3D from eight stationary cameras placed around the scene. We then project the skeleton joints into the fisheye images of our head-mounted camera rig. The projection requires tracking the rigid motion of our head-mounted camera rig relative to the stationary cameras of the motion-capture system, for which we use a large checkerboard rigidly attached to our camera rig (Figure 6.3). We detect the checkerboard in all stationary cameras in which it is visible, and triangulate the 3D positions of its corners to estimate the pose and orientation of the camera rig. Using Scaramuzza *et al.*'s camera distortion model, we then project the 3D joint locations into the fisheye images recorded by our camera rig.

Dataset Augmentation We record video sequences of eight subjects performing various motions in a green-screen studio. For the training set, we replace the

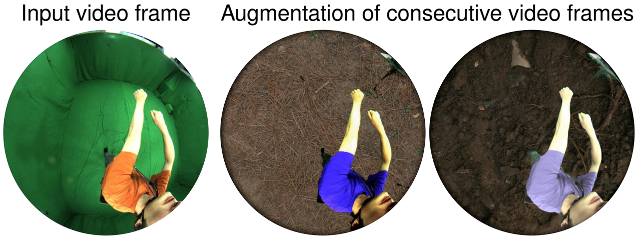


Figure 6.4: Illustration of our dataset augmentation using randomized backgrounds, intrinsic recoloring and gamma jittering. Note the varied shirt colors as well as brightness of the trousers and skin, which help prevent overtraining of the CNN-based joint detector.

background of each video frame, using chroma keying, with a random, floor-related image from Flickr, as our fisheye cameras mostly see the ground below the tracked subject. Please note that training with real backgrounds could give the CNN additional context, but is prone to overfitting to a (necessarily) small set of recorded real backgrounds. In addition, we augment the appearance of subjects by varying the colors of clothing, while preserving shading effects, using intrinsic recoloring Meka *et al.* (2016). This is, to our knowledge, the first application of intrinsic recoloring for augmenting datasets. We also apply a random gamma curve ($\gamma \in [0.5, 2]$) to simulate changing lighting conditions. We furthermore exploit the shared plane of symmetry of our camera rig and the human body to train a single detector on a dataset twice the size by mirroring the images and joint-location annotations of the right-hand camera to match those of the left-hand camera during training, and vice versa during motion capture. Thanks to the augmentation, both background and clothing colors are different for every frame (see Figure 6.4), which prevents overfitting to the limited variety of the captured appearances. This results in a training set of six subjects and ~75,000 annotated fisheye images. Two additional subjects are captured and prepared for validation purposes.

6.4.3.2 *Detector Learning*

Our starting point for learning an egocentric body-part detector for fisheye images is the 101-layer residual network (He *et al.*, 2016) trained on the MPII Human Pose dataset (Andriluka *et al.*, 2014), which contains ~19,000 internet images that were manually annotated in a crowd-sourced effort, and the Leeds Sports Extended dataset (Johnson and Everingham, 2011) of 10,000 images. We remove the original prediction layers and replace them with ones that output 18 body-part heat maps[1]. The input video frames are scaled to a resolution of 640×512 pixels, the predicted heat maps are of 8× coarser resolution. We then fine-tune the CNN on our fisheye dataset for 220,000 iterations with a learning rate of 0.002, and drop it to 0.0002 for 20,000 additional iterations. The number of training iterations is chosen based on

[1] We jointly learn heat maps for the head and neck, plus the left and right shoulders, elbows, wrists, hands, hips, knees, ankles and feet.

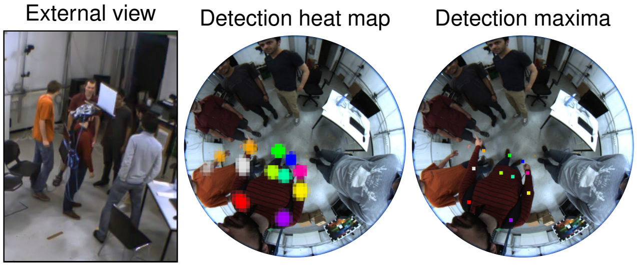


Figure 6.5: Color-coded joint-location detections on the Crowded sequence. For crowded scenes (left), detections can be multi-modal (center). However, the maximum (right) lies on the user. We exclude knee, hand and ankle locations for clearer visualization.

performance on the validation set. We randomly scale images during training by up to ±15% to be more robust to variations in user size. Figure 6.5 (center) visualizes the computed heat maps for selected body parts. We demonstrate generalization capability to a large variety of backgrounds, changing illumination and clothing colors in Section 6.5.3.

6.4.3.3 *Body-Part Detection Energy*

Inspired by Elhayek *et al.* (2015), who exploit detections in outside-in motion capture, we integrate the learned detections, in the form of heat maps as shown in Figure 6.5, into the objective energy (Equation 6.1) as a soft constraint. For each detection label, the location with maximum confidence, $(\hat{u}, \hat{v})$, is selected and an associated 3D Gaussian is attached to the corresponding skeleton body part. This association can be thought of as giving a distinct color to each body-part label. The Gaussian is used to compute the spatial agreement of the detection and body-part location in the same way as in the color similarity E_{color}, only the color distance $d(\cdot, \cdot)$ in Equation 6.5 is replaced with the predicted detection confidence at $(\hat{u}, \hat{v})$. For instance, a light green Gaussian is placed at the right knee and is associated with the light green knee detection heat map at $(\hat{u}, \hat{v})$, then their agreement is maximal when the Gaussian's center projects on $(\hat{u}, \hat{v})$. By this definition, $E_{\text{detection}}$ forms the sum over the detection agreements of all body parts and in both cameras. We weight its influence by $\lambda_{\text{detection}} = 1/3$.

6.4.4 Real-Time Optimization

The volumetric ray-casting method of Rhodin *et al.* (2015) models occlusion as a smooth phenomenon by integrating the visibility computations within the objective function instead of applying a depth test once before optimization. While this is beneficial for optimizing disocclusions, it introduces dense pairwise dependencies between all Gaussians: the visibility $\mathcal{V}_q$ (Equation 6.4) of a single Gaussian can be evaluated in linear time in terms of the number of Gaussians, N_q, but E_{color} – and its gradient with respect to all Gaussians – has quadratic complexity in N_q.

To nevertheless reach real-time performance, we introduce a new parallel stochastic optimization approach. The ray-casting formulation allows a natural parallelization of $E_{\text{detection}}$ and E_{color} terms and their gradient computation across pixels (u, v) and Gaussians G_q. We also introduce a traversal step, which determines the Gaussians that are close to each ray, and excludes distant Gaussians with negligible contribution to the energy. These optimizations lead to significant run-time improvements, particularly when executed on a GPU, but only enable interactive frame rates.

We achieve further reductions in run times by introducing a statistical optimization approach that is tailored to the ray-casting framework. The input image pixels are statistically sampled for each gradient iteration step, as proposed by Blanz and Vetter (1999). In addition, we sample the volumetric body model by excluding Gaussians from the gradient computation at random, individually for each pixel, which improves the optimization time to 10 fps and more.

6.5 EVALUATION

6.5.1 Hardware Prototypes

We show the two EgoCap prototypes used in this work in Figure 6.1 (left). *EgoRig1* consists of two fisheye cameras attached to a standard bike helmet. It is robust and well-suited for capturing outdoor activities and sports. *EgoRig2* builds on a lightweight wooden rig that holds two consumer cameras and is glued to an Oculus VR headset. It weighs only 65 grams and adds minimal discomfort on the user. Both prototypes are equipped with 180° fisheye lenses and record with a resolution of 1280×1024 pixels at 30 Hz. Note that the checkerboard attached to *EgoRig1* in several images is not used for tracking (only used in training and validation dataset recordings).

Body-Part Visibility For egocentric tracking of unconstrained motions, the full 180° field of view is essential for egocentric tracking. We evaluate the visibility of selected body parts from our egocentric rig with different (virtual) field-of-view angles in Figure 6.6. Only at 180 degrees are almost all body parts captured, otherwise even small motions of the head can cause the hand to leave the recording volume. The limited field of view of existing active depth sensors of 60–80 degrees restricts their applicability to egocentric motion capture in addition to their higher energy consumption and interference with other light sources.

6.5.2 Runtime

For most tracking results, we use a resolution of 128×128 pixels and 200 gradient-descent iterations. Our CPU implementation runs at ten seconds per frame on a

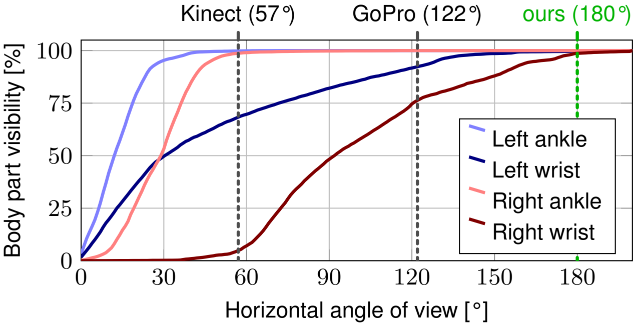


Figure 6.6: Visibility of selected body parts for different camera angles of view, for the left-hand camera in our rig over a 5-minute recording. Seeing the right wrist 95 percent of the time requires an angle of view in excess of 160°, which is only practical with fisheye lenses.

Xeon E5-1620 3.6 GHz, which is similar to run times reported by **?**. Straightforward parallelization on the GPU reduces run times to two seconds per frame. The body-part detector runs on a separate machine, and processes 6 images per second on an Nvidia Titan GPU and a Xeon E5-2643 3.30 GHz.

For some experiments (see 6.6.3), we use a resolution of 120×100 pixels and enable stochastic optimization. Then, purely color-based optimization reaches 10 to 15 fps for 50 gradient iterations (2–3 ms per iteration), i.e. close to real-time performance. Our body-part detector is not optimized for speed and cannot yet run at this frame rate, but its implementation could be optimized for real-time processing, so a real-time end-to-end approach would be feasible without algorithmic changes.

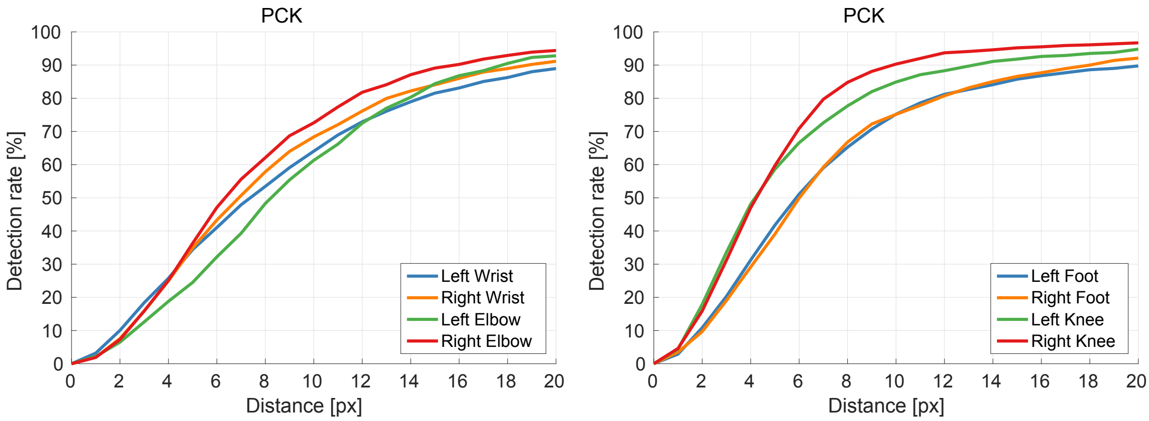
6.5.3 Body-Part Detections

We first evaluate the learned body-part detectors, irrespective of generative components, using the percentage of correct keypoints (PCK) metric (Sapp and Taskar, 2013; Tompson *et al.*, 2014). We evaluate on a validation set, Validation2D, of 1000 images from a 30,000-frame sequence of two subjects that are not part of the training set and wear dissimilar clothing. Validation2D is augmented with random backgrounds using the same procedure as for the training set, such that the difficulty of the detection task matches the real-world sequences. We further validated that overfitting to augmentation is minimal, by testing on green-screen background, with equivalent results.

Dataset Augmentations 6.1 presents the evaluation of proposed data augmentation strategies. Background augmentation during training brings a clear improvement. It provides a variety of challenging negative samples for the training of the detector, which is of high importance. Secondly, the performance is further boosted by employing intrinsic video for cloth recoloring, which additionally increases the diversity of training samples. The improvement of about two percent is consistent

Training dataset setting	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	PCK	AUC
green-screen background	75.5	46.8	18.8	13.6	17.4	7.2	4.5	22.4	10.0
+ background augmentation	84.7	87.5	90.9	89.1	97.7	94.2	86.4	89.5	56.9
+ intrinsic recoloring	**86.2**	**96.1**	**93.6**	**90.1**	**99.1**	**95.8**	**90.9**	**92.5**	**59.4**

Table 6.1: Part detection accuracy in terms of the percentage of correct keypoints (PCK) on the validation dataset Validation2D of 1000 images, evaluated at 20 pixel threshold for three CNNs trained with different data augmentation strategies (6.4.3.1). AUC is area under curve evaluated for all thresholds up to 20 pixels.



(a) Arm joints

(b) Leg joints

Figure 6.7: Pose estimation results in terms of percentage of correct keypoints (PCK) for different distance thresholds on Validation2D.

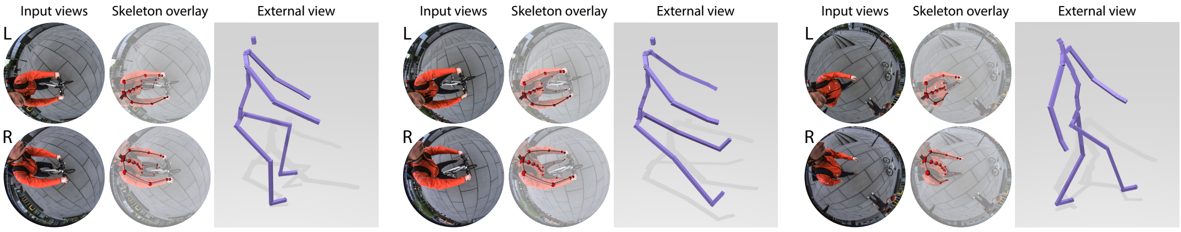


Figure 6.8: EgoCap enables outdoor motion capture with virtually unconstrained extent. Full-body pose is accurately estimated for fast Biking (left and center) and for unconstrained Walk (right). The model is tailored to handle the present occlusions and strong image distortion.

across all body parts.

Detection Accuracy Figure 6.7 contains the plots of PCK at different distance thresholds for arms and legs evaluated on sequence Validation2D. We achieve high accuracy, with slightly lower detection reliability of terminal limbs (wrists, feet). This can either be due to more articulation or, in case of the feet, due to higher occlusion by knees and their small appearance due to the strong fisheye distortion. The 2D detection accuracy of feet and wrists is comparable, even though feet are further away, and similar pixel error hence translates to larger 3D errors, as evaluated in the next section. We additionally evaluated the training set size. We found that subject variation is important: using only three out of six subjects, the PCK performance dropped by 2.5 percent points. Moreover, using a random subset of 10% of the original database size reduces the PCK by 2 points, i.e. using more than three frames per second is beneficial. Using a 50% subset did not degrade performance, showing that consecutive frames are not crucial for our per-frame model, but could be beneficial for future research, such as for temporal models.

6.5.4 3D Body Pose Accuracy

Our main objective is to infer 3D human pose from the egocentric views, despite occlusions and strong fisheye image distortions. We quantitatively evaluate the 3D body pose accuracy of our approach on two sequences, ValidationWalk and ValidationGest. Ground-truth data is obtained with the Captury Studio, a state-of-the-art marker-less commercial multi-view solution with eight video cameras and 1–2 cm accuracy. The two systems are used simultaneously and their relative transformation is estimated with a reference checkerboard, see Figure 6.3. We experimented with raw green-screen and with randomly replaced background. Error values are estimated as the average Euclidean 3D distance over 17 joints, including all joints with detection labels, except the head. Reconstructions on green and replaced backgrounds are both 7±1 cm for a challenging 250-frame walking sequence with occlusions, and 7±1 cm on a long sequence of 750 frames of gesturing and interaction. During gesturing, where arms are close to the camera, upper body (shoulder, elbow, wrist, finger) joint accuracy is higher than for the lower body (hip, knee, ankle, and toe) with 6 cm and 8 cm average error, respectively. During walking, upper and lower body error is similar with 7 cm. Please note that slight differences in skeleton topology between ground truth and EgoCap exist, which might bias the errors.

Despite the difficult viewing angle and image distortion of our egocentric setup, the overall 3D reconstruction error is comparable to state-of-the-art results of outside-in approaches (Rhodin *et al.*, 2015; Elhayek *et al.*, 2015; Amin *et al.*, 2009; Sigal *et al.*, 2010; Belagiannis *et al.*, 2014), which reach 5–7 cm accuracy from two or more cameras, but only in small and open recording volumes, and for static cameras. In contrast, our algorithm scales to very narrow and cluttered scenes (see Figure 6.9) as well as to wide unconstrained performances (see Figure 6.8). No existing algorithm is directly applicable to these conditions and the strong distortions of the fisheye cameras, precluding a direct comparison. Closest to our approach is the fundamentally offline inside-out method of Shiratori *et al.* (2011), who use 16 body-worn cameras facing outwards, reporting a mean joint position error of 2 cm on a slowly performed indoor walking sequence. Visually, their outdoor results show similar quality to our reconstructions, although we require fewer cameras, and can handle crowded scenes. It depends on the application whether head gear or body-worn cameras less impair the user's performance.

6.5.5 Model Components

Our objective energy consists of detection, color, smoothness, and pose prior terms. Disabling the smoothness term increases the reconstruction error on the validation sequences by 3 cm. Without the color term, accuracy is reduced by 0.5 cm. We demonstrate in the supplemental video that the influence of the color term is more significant in the outdoor sequences for motions that are very dissimilar to the training set. Disabling the detection term removes the ability to recover from tracking failures, which are usually unavoidable for fully automatic motion capture of long sequences with challenging motions. High-frequency noise is filtered with a Gaussian low-pass filter of window size 5.

6.6 APPLICATIONS

We further evaluate our approach in three application scenarios with seven sequences of lengths of up to 1500 frames using *EgoRig1*, in addition to the three quantitative evaluation sequences. The captured users wear clothes not present in the training set.

6.6.1 Unconstrained/Large-Scale Motion Capture

We captured a Basketball sequence outdoors, which shows quick motions, large steps on a steep staircase, and close interaction of arms, legs and the basketball

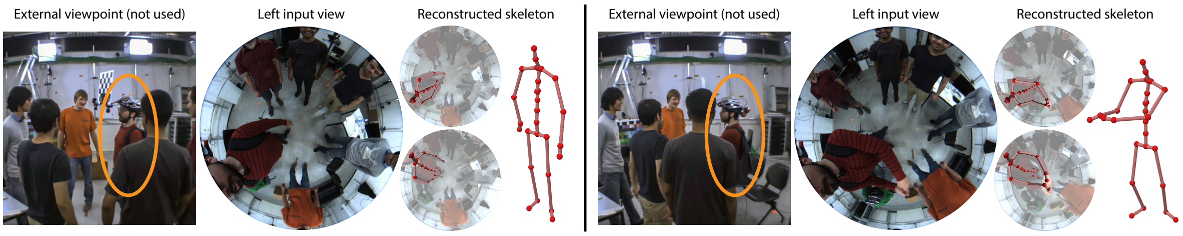


Figure 6.9: Capturing social interaction in crowded scenes is of importance, but occlusions pose difficulties for existing outside-in approaches (left). The egocentric view enables 3D pose estimation, as demonstrated on the Crowded sequence. The visible checkerboard is not used.

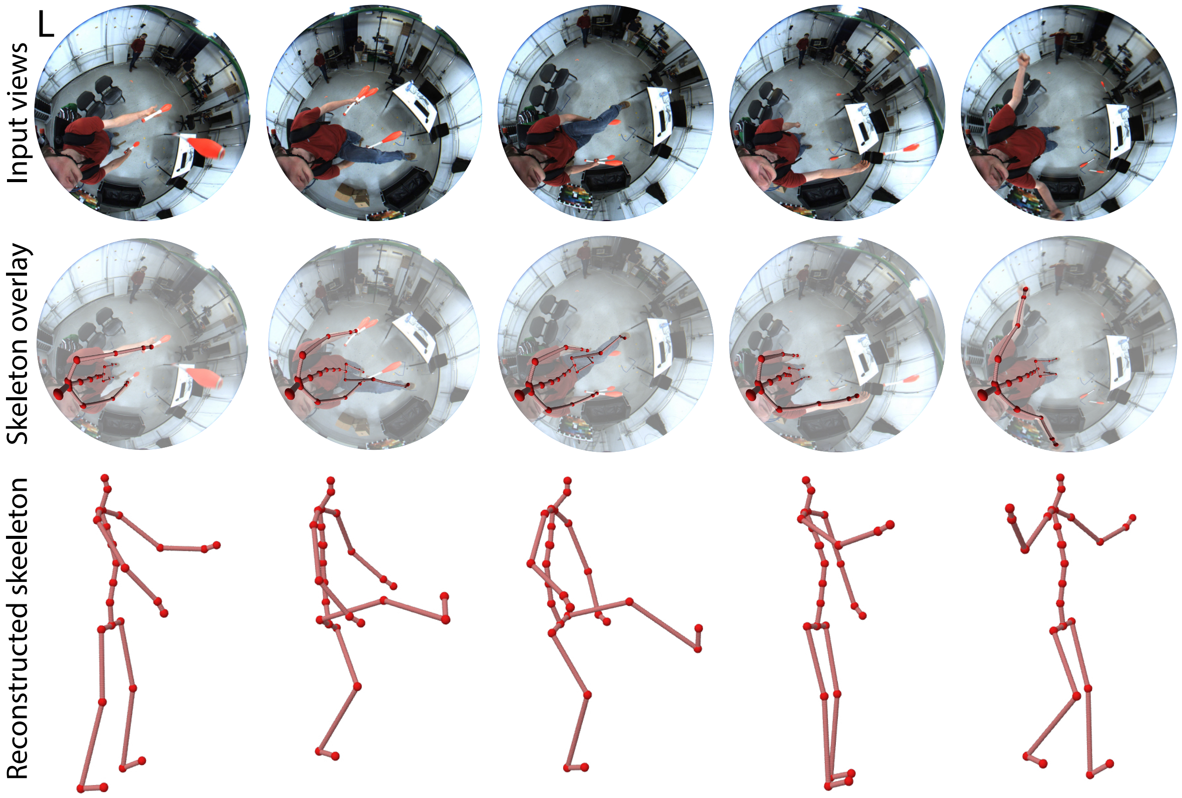


Figure 6.10: Reconstruction results on the Juggler sequence, showing one input view and the estimated skeleton. Despite frequent self-occlusions, our approach robustly recovers the skeleton motion.

(supplemental video). We also recorded an outdoor Walk sequence with frequent arm-leg self-occlusions (Figure 6.8, right). With EgoCap, a user can even motion capture themselves while riding a bike in a larger volume of space (Bike sequence, Figure 6.8, left and center). The pedaling motion of the legs is nicely captured, despite frequent self-occlusions; the steering motion of the arms and the torso is also reconstructed. Even for very fast absolute motions, like this one on a bike, our egocentric rig with cameras attached to the body leads to little motion blur, which challenges outside-in optical systems. All this would have been difficult with alternative motion-capture approaches.

Note that our outdoor sequences also show the resilience of our method to different appearance and lighting conditions, as well as the generalization of our detector to a large range of scenes.

6.6.2 Constrained/Crowded Spaces

We also tested EgoCap with *EgoRig1* for motion capture on the Crowded sequence, where many spectators are interacting and occluding the tracked user from the outside (Figure 6.9). In such a setting, as well as in settings with many obstacles and narrow sections, outside-in motion capture, even with a dense camera system, would be difficult. In contrast, EgoCap captures the skeletal motion of the user in the center with only two head-mounted cameras.

The egocentric camera placement is well-suited for capturing human-object interactions too, such as the juggling performance Juggler (Figure 6.10). Fast throwing motions as well as occlusions are handled well. The central camera placement ensures that objects that are manipulated by the user are always in view.

6.6.3 Tracking for Immersive VR

We also performed an experiment to show how EgoCap could be used in immersive virtual reality (VR) applications. To this end, we use *EgoRig2* attached to an Oculus VR headset and track the motion of a user wearing it. We build a real-time demo application running at up to 15 fps, showing that real-time performance is feasible with additional improvements on currently unoptimized code. In this Live test, we only use color-based tracking of the upper body, without detections, as the detector code is not yet optimized for speed. The Live sequence shows that body motions are tracked well, and that with such an even more lightweight capture rig, geared for HMD-based VR, egocentric motion capture is feasible. In the supplemental video, we show an additional application sequence 'VR', in which the the user can look down at their virtual self while sitting down on a virtual sofa. Current HMD-based systems only track the pose of the display; our approach adds motion capture of the wearer's full body, which enables a much higher level of immersion.

Global Pose Estimation For free roaming, the global rig pose can be tracked independently of external devices using structure-from-motion in the fisheye views. We demonstrate combined local and global pose estimation on the Biking, Walk, and VR sequence, using the structure-from-motion implementation of Moulon *et al.* (2013) provided in the OpenMVG library, see Figure 6.11 and the accompanying video. Such complete motion capture paves the way for immersive roaming in a fully virtual 3D environment.

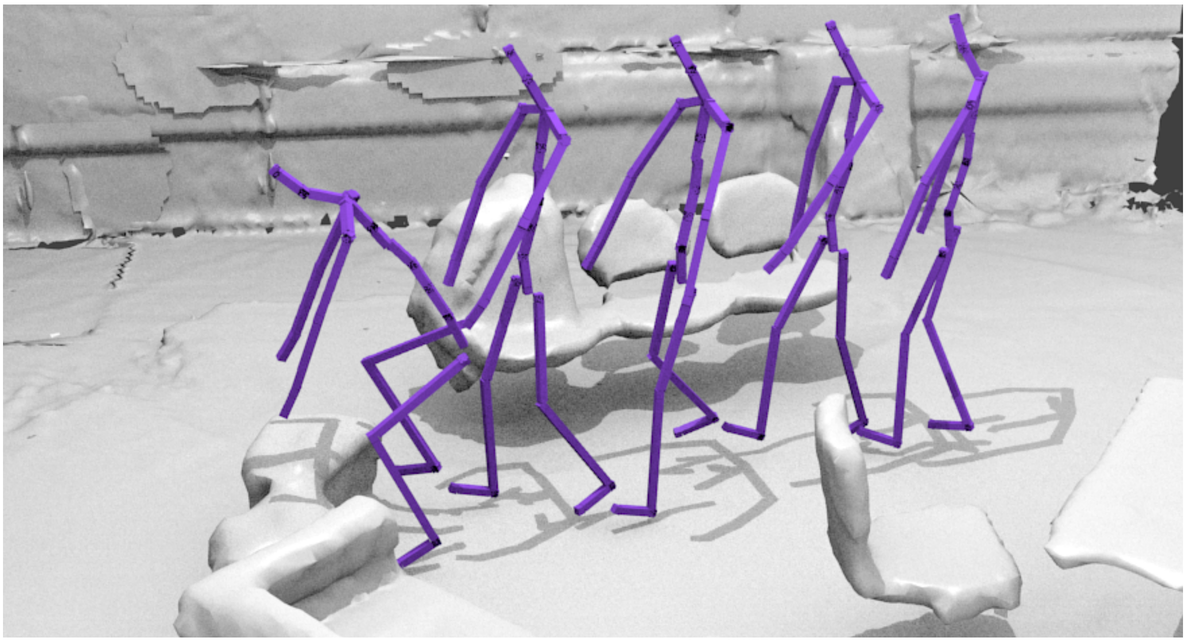


Figure 6.11: Complete motion-capture example VR, in which our egocentric pose tracking is combined with global pose tracking using structure-from-motion, shown as a motion sequence in a 3D reconstruction of the scene. In a VR scenario, this would allow free roaming and interaction with virtual objects.

6.7 DISCUSSION AND LIMITATIONS

We developed the first stereo egocentric motion-capture approach for indoor and outdoor scenes, that also works well for very crowded scenes. The combination of generative and detection-based pose estimation make it fare well even under poses with notable self-occlusions. Similar to other outside-in optical methods, tracking under occlusions by objects in the environment, such as a table, may lead to tracking failures. However, the detections enable our tracker to quickly recover from such occlusion failures. Interestingly, the egocentric fisheye camera setup provides stronger perspective cues for motion towards and away from the camera than with normal optics. The perspective effect of the same motion increases with proximity to the camera. For instance, bending an arm is a subtle motion when observed from an external camera, but when observed in proximity, the same absolute motion causes large relative motion, manifesting in large displacements and scaling of the object in motion.

The algorithm in this chapter focuses on an entirely new way of capturing the full egocentric skeletal body pose, that is decoupled from global pose and rotation relative to the environment. Global pose can be inferred separately by structure-from-motion from the fisheye cameras or is provided by HMD tracking in VR applications. Fisheye cameras keep the whole body in view, but cause distortions reducing the image resolution of distant body parts such as the legs. Therefore, tracking accuracy of the upper body is slightly higher than that of the lower body. Also, while overall tracking accuracy of our research prototype is still lower than with commercial outside-in methods, it shows a new path towards more unconstrained capture in the future. Currently, we have no real-time end-to-end prototype. We are confident that this would be feasible without algorithm redesign, yet felt that real-time performance is not essential to demonstrate the algorithm and its general feasibility.

Our current prototype systems may still be a bit bulky, but much stronger miniaturization becomes feasible in mass production; the design of *EgoRig2* shows this possibility. Some camera extension is required for lower-body tracking and might pose a problem with respect to social acceptance and ergonomics for some applications; However, we did not encounter practical issues during our recordings and VR tests, as users naturally keep the area in front of their head clear to not impair their vision. Moreover, handling changing illumination is still an open problem for motion capture in general and is not the focus of our work. For dynamic illumination, the color model would need to be extended. However, the CNN performs one-shot estimation and does not suffer from illumination changes. The training data also contains shadowing from the studio illumination, although extreme directional light might still cause inaccuracies. Additionally, loose clothing, such as a skirt, is not part of the training dataset and hence likely to reduce pose accuracy.

6.8 CONCLUSION

This chapter presented EgoCap, the first approach for marker-less egocentric full-body motion capture with a head-mounted fisheye stereo rig. It is based on a pose optimization approach that jointly employs two components. The first is a new generative pose estimation approach based on a ray-casting image formation model enabling an analytically differentiable alignment energy and visibility model. The second component is a new CNN-based body-part detector for fisheye cameras that was trained on the first automatically annotated real-image training dataset of egocentric fisheye body poses. EgoCap's lightweight on-body capture strategy bears many advantages over other motion-capture methods. It enables motion capture of dense and crowded scenes, and reconstruction of large-scale activities that would not fit into the constrained recording volumes of outside-in motion-capture methods. It requires far less instrumentation than suit-based or exoskeleton-based approaches. EgoCap is particularly suited for HMD-based VR applications; two cameras attached to an HMD enable full-body pose reconstruction of your own virtual body to pave the way for immersive VR experiences and interactions.

LEARNING 3D OBJECT SHAPE AND CAMERA POSE 7

Contents

This chapter addresses the problem of learning accurate 3D shape and camera pose from a collection of unlabeled category-specific images. It describes an approach consisting of a convolutional network trained to predict both the shape and the pose from a single image by minimizing the reprojection error: given several views of an object, the projections of the predicted shapes to the predicted camera poses should match the provided views. In order to deal with ambiguity of the camera pose, we introduce an ensemble of pose predictors which we then distill to a single "student" model. To allow for efficient learning of high-fidelity shapes, we represent the shapes by point clouds and devise a formulation allowing for differentiable projection of these. The experiments show that the distilled ensemble of pose predictors learns to estimate the pose accurately, while the point cloud representation allows to predict detailed shape models.

7.1 INTRODUCTION

We live in a three-dimensional world, and a proper understanding of its volumetric structure is crucial for acting and planning. However, we perceive the world mainly via its two-dimensional projections. Based on these projections, we are able to infer the three-dimensional shapes and poses of the surrounding objects. How does this volumetric shape perception emerge from observing only from two-dimensional projections? Is it possible to design learning systems with similar capabilities?

Deep learning methods have recently shown promise in addressing these questions (Yan *et al.*, 2016; Tulsiani *et al.*, 2017c). Given a set of views of an object and the corresponding camera poses, these methods learn 3D shape via the reprojection error: given an estimated shape, one can project it to the known camera views and compare to the provided images. The discrepancy between these generated projections and the training samples provides training signal for improving the shape estimate. Existing methods of this type have two general restrictions. First, these approaches assume that the camera poses are known precisely for all provided images. This is a practically and biologically unrealistic assumption: a typical intelligent agent only has access to its observations, not its precise location relative to objects in the world. Second, the shape is predicted as a low-resolution (usually 32^3 voxels) voxelated volume. This representation can only describe very rough shape of an object. It should be possible to learn finer shape details from 2D supervision.

In this chapter we discuss an algorithm capable of learning high-fidelity shape models solely from their projections, without ground truth camera poses. This setup is challenging for two reasons. First, estimating both shape and pose is a chicken-and-egg problem: without a good shape estimate it is impossible to learn accurate pose because the projections would be uninformative, and vice versa, an accurate pose estimate is necessary to learn the shape. Second, pose estimation is prone to local minima caused by ambiguity: an object may look similar from two viewpoints, and if the network converges to predicting only one of these in all cases, it will not be able to learn predicting the other one. We find that the first problem can be solved surprisingly well by joint optimization of shape and pose predictors: in practice, good shape estimates can be learned even with relatively noisy pose predictions. The second problem, however, leads to drastic errors in pose estimation. To address this, we train a diverse ensemble of pose predictors and distill those to a single student model.

To allow learning of high-fidelity shapes, we use the point cloud representation, in contrast with voxels used in previous works. Point clouds allow for computationally efficient processing, can produce high-quality shape models (Fan *et al.*, 2017), and are conceptually attractive because they can be seen as "matter-centric", as opposed to "space-centric" voxel grids. To enable learning point clouds without explicit 3D supervision, we implement a differentiable projection operator that, given a point set and a camera pose, generates a 2D projection – a silhouette, a color image, or a depth map. We dub the formulation "Differentiable Point Clouds".

We evaluate the proposed approach on the task of estimating the shape and the camera pose from a single image of an object. The method successfully learns to predict both the shape and the pose, with only a minor performance drop relative to a model trained with ground truth camera poses. The point-cloud-based formulation allows for effective learning of high-fidelity shape models when provided with images of sufficiently high resolution as supervision. We demonstrate learning point clouds from silhouettes and augmenting those with color if color images are available during training. Finally, we show how the point cloud representation allows to automatically discover semantic correspondences between objects.

7.2 RELATED WORK

Reconstruction of three-dimensional shapes from their two-dimensional projections has a long history in computer vision, constituting the field of 3D reconstruction. A review of this field goes outside of the scope of our work; however, we briefly list several related methods. Cashman and Fitzgibbon (2013) use silhouettes and keypoint annotation to reconstruct deformable shape models from small class-specific image collections, Vicente *et al.* (2014) apply similar methods to a large-scale Pascal VOC dataset, Tulsiani *et al.* (2017a) reduce required supervision by leveraging computer vision techniques. These methods show impressive results even in the small data regime; however, they have difficulties with representing diverse and complex shapes. Loper and Black (2014) implement a differentiable renderer and apply it for analysis-by-synthesis. Our work is similar in spirit, but operates on point clouds and integrates the idea of differentiable rendering with deep learning. The approach of Rhodin *et al.* (2015) is similar to our technically in that it models human body with a set of Gaussian density functions and renders them using a physics-motivated equation for light transport. Unlike in our approach, the representation is not integrated into the learning framework and requires careful initial placement of the Gaussians, making it unsuitable for automated reconstruction of arbitrary shape categories. Moreover, the projection method scales quadratically with the number of Gaussians, which limits the maximum fidelity of the shapes being represented.

Recently the task of learning 3D structure from 2D supervision is being addressed with deep-learning-based methods. The methods are typically based on reprojection error – comparing 2D projections of a predicted 3D shape to the ground truth 2D projections. Yan *et al.* (2016) learn 3D shape from silhouettes, via a projection operation based on selecting the maximum occupancy value along a ray. Tulsiani *et al.* (2017c) devise a differentiable formulation based on ray collision probabilities and apply it to learning from silhouettes, depth maps, color images, and semantic segmentation maps. Lin *et al.* (2018) represent point clouds by depth maps and re-project them using a high resolution grid and inverse depth max-pooling. Concurrently with us, Kato *et al.* (2018) propose a differentiable renderer for meshes and use it for learning mesh-based representations of object shapes. All these methods require exact ground truth camera pose corresponding to the 2D projections used for training. In contrast, we aim to relax this unrealistic assumption and learn only from the projections.

Rezende *et al.* (2016) explore several approaches to generative modeling of 3D shapes based on their 2D views. One of the approaches does not require the knowledge of ground truth camera pose; however, it is only demonstrated on a simple dataset of textured geometric primitives. Most related to our submission is the concurrent work of Tulsiani *et al.* (2018). The work extends the Differentiable Ray Consistency formulation Tulsiani *et al.* (2017c) to learning without pose supervision. The method is voxel-based and deals with the complications of unsupervised pose learning using reinforcement learning and a GAN-based prior. In contrast, we make use of a point cloud representation, use an ensemble to predict the pose, and do not require a prior on the camera poses.

7.2.1 3D shape Representations.

Voxel grids. The issue of representation is central to deep learning with volumetric data. The most commonly used structure is a voxel grid - a direct 3D counterpart of a 2D pixelated image (Choy *et al.*, 2016; Wu *et al.*, 2016). This similarity allows for simple transfer of convolutional network architectures from 2D to 3D. However, on the downside, the voxel grid representation scales cubically with the resolution and leads to memory- and computation-hungry architectures. Most works therefore limit voxel grids to the size of 32^3 or 64^3 which is insufficient to represent detailed shapes. A potential solution is to use voxel grids with adaptive subdivision strategy such as octrees (Tatarchenko *et al.*, 2017), however this requires a non-trivial implementations.
Polygon meshes. 3D polygon meshes are another attractive representation and provide explicit information about object surface. Moreover, meshes are a compact representation able to represent large flat surfaces with only a handful of triangles. Kato *et al.* (2018) train a mesh-generating network such that the silhouettes of the predicted mesh match ground truth silhouettes. The key component is a neural mesh renderer that approximates a gradient of a normally discrete rasterization operation by smoothing. In order to generate vertex coordinates of a mesh, Kato *et al.* (2018) start with an isotropic sphere and train a network to predict an additive deformation of its vertices. The main limitation of such an approach is in its inability to represent shapes that are not homeomorphic to a sphere (i.e. shapes with holes). Similarly, the FoldNet by Yang *et al.* (2018) generates point clouds by iteratively deforming a 2D grid of points. The AtlasNet by Groueix *et al.* (2018) can represent arbitrary topologies by learning several MLPs each of which deforms a 2D grid to a surface patch. The final shape is then computed as the union of these patches. On the downside, this may result in self-intersection of several mesh faces.

Part-based models. Representing object shapes as collection of parts is another popular paradigm and has a long tradition in computer vision. Decomposing objects into a small number of shape preimitives results in a compact low-dimensional representations. Tulsiani *et al.* (2017b) learn to approximate the shapes with a set of cuboid primitives with the neural network predicting their transformation parameters. The network is trained in an unsupervised way and discoveres assignment of shape primitives without any part annotations in the dataset. Zou *et al.* (2017) develop this idea further by training a sequential generator of shape primitives based on RNNs; Niu *et al.* (2018) train a network that predicts a hierarchy of cuboid primitives with mutual relationships between them. Li *et al.* (2017) train a part-based autoencoder that produces a hierarchy of parts. A separate network is trained to generate a volumetric representation of geometry for each part bounding box. However the approach requires supervision for hierarchical representations. Paschalidou *et al.* (2020) learn a hierarchy of parts represented by a binary tree of superquadric primitives (Paschalidou *et al.*, 2019) without part-level supervision with only watertight meshes available for training.

Multi-view depth maps. A popular and efficient representation of 3D shapes is multi-view depth maps. Soltani *et al.* (2017) predict silhouettes and depth maps for a predefined set of views. The final shape is given by a union of depth maps projected to the 3D space and a refinement step is applied filtering out points not consistent with multi-view silhouettes. An advantage of such an approach is that one can reuse the machinery of convolutional networks for 2D images and apply it for prediction of depth map achieving higher resolutions than 3D voxels. Matryoshka Networks (Richter and Roth, 2018) predict six axis-aligned opposite depth maps corresponding to the unit cube and the final shape is taken as an intersection of occupancy grids imposed by the depth maps. In order to reconstruct details occluded from all six views Richter and Roth (2018) build a shape from L shape layers by iteratively adding and subtracting shapes. Wu *et al.* (2018) represent objects as a 3D skeleton (keypoints and connections between them) and learn to infer 3D skeletons via a two-stage inference: 2D keypoint localization and subsequent 3D lifting.

Point clouds. Another popular representation of 3D geometry are point clouds. Generating point clouds for single view 3D reconstruction was pioneered by Fan *et al.* (2017). The authors propose to use Chamfer Distance and Earth Mover's Distance, two distance metrics defined on point clouds, for supervision of point set generating networks. Lin *et al.* (2018) predict multi-view depth maps to generate point clouds, similarly to the work by Soltani *et al.* (2017). We choose to use point clouds in this work, since they are less overcomplete than voxel grids and allow for effective networks architectures, but at the same time are more flexible than mesh-based or skeleton-based representations. The downside of point-based representation is that individual points do not provide information about the surface and connectivity which poses a challenge during reconstruction.

Implicit functions. Since the publication of our work in 2018 a new and powerful class of neural representations of 3D shapes had emerged, known as Occupancy Networks (Mescheder *et al.*, 2019), DeepSDF (Park *et al.*, 2019) and Implicit Fields (Chen and Zhang, 2019). Instead of providing occupancy at fixed discrete locations (voxel grids) they represent shapes as a function that maps a given 3D coordinate to an occupancy value: $\phi : \mathbb{R}^3 \to \{0, 1\}$. The function is effectively a classifier that predicts whether a point is inside or outside the object. The zero level-set of ϕ corresponds to the surface of the object and could be extracted with the marching cubes algorithm (Lorensen and Cline, 1987). Because the space in which the object lives is not stored explicitly, this class of shape representations is commonly referred to as implicit functions. They are implemented as neural networks that take the shape's latent code and 3D coordinate query as inputs and return either an occupancy value (Mescheder *et al.*; Chen and Zhang) or a signed distance to the surface (SDF) (Park *et al.*). The output of such a network can be sampled at resolutions higher than that of the training shapes. Normals to the surface can be estimated as gradients of the implicit function wrt. the input coordinates by backpropagation. Implicit representation can support different types of downstream tasks such as shape modeling with auto-encoders, single image 3D reconstruction or point cloud completion. While Mescheder *et al.* train a variational auto-encoder for shape modeling, Park *et al.* propose a novel encoder-less architecture which they name auto-decoder. Latent codes are optimized jointly with the parameters of the decoder by gradient descent instead of being obtained with an encoder network.

Scaling implicit functions. In the original formulation implicit functions are conditioned only on the global shape code and an input coordinate and struggle to model global configurations of complex shapes. Saito *et al.* (2019) introduce Pixel-aligned Implicit Functions (PIFu) for single image 3D reconstruction. They learn per-pixel feature vectors with a fully convolutional encoder. Then, given such per-pixel feature vector and a depth value z an implicit function is trained to classify whether the corresponding 3D point is inside or outside the surface. Predicting pixel-conditioned occupancy values allows to model high-frequency details which is crucial for reconstructing humans with arbitrary body types, clothing and complex hairstyles. DISN by Xu *et al.* (2019a) utilise local information by projecting a 3D query point to an image plane with an estimated viewpoint parameters and extract local features around projected 2D location. The local features supplement the global latent code when evaluating the implicit function and allow for significantly more detailed reconstructions. The key difference to PIFu is that the latter performs reconstruction in the camera coordinate frame and does not require viewpoint estimation. Chibane *et al.* (2020) apply implicit functions for reconstruction from 3D input modalities such as point clouds (partial and complete) or voxel grids (low- or high-resolution). This work essentially extends PIFu to the 3D domain: instead of representing the shape with a single feature vector it extracts a 3D tensor of features with a fully convolutional 3D U-Net (Ronneberger *et al.*, 2015); the authors name their method Implicit Feature Networks (IF-Nets). Differently to the prior methods, the implicit function is evaluated on feature vectors sampled with trilinear interpolation instead of feeding the query coordinate as input, which improves the locality of predictions. Convolutional Occupancy Networks by Peng *et al.* (2020) also utilize U-Nets to produce feature tensors and evaluate implicit functions locally. Differently, they propose several different ways of 3D encoding, including 2D U-Nets on orthogonal 2D projections of 3D features and evaluate their method on larger scale scene-level reconstructions.

Structured Implicit Functions (SIFs) of Genova *et al.* (2019) model an implicit function as an additive mixture of a fixed number of shape templates. Each shape template is represented by an axis aligned 3D Gaussian (its parameters are predicted by a network) and the occupancy function of the whole shape is a weighted sum of shape templates. Experiments demonstrate interpretability of template based representation where the same templates tend to represent the same semantic parts, as well as clustering by template parameters tends to group the same semantic classes together. At the same time the strengths of implicit functions are retained. Local Deep Implicit Functions (LDIF) of Genova *et al.* (2020) extend SIFs by learning an additional occupancy network per shape template. While Gaussian-based templates are responsible for modeling global shape configuration, occupancy networks operating in a local coordinate system of a template can represent fine-grained details. CvxNets of Deng *et al.* (2020) represent convex shapes as intersections of half spaces induced by hyperplanes, with an auto-encoder predicting the parameters of the hyperplanes. Non-convex shapes are approximated as a union of convexes. CvxNets unify implicit and explicit representations, since an explicit representation can always be obtained by extracting the mesh as an intersection of the hyperplanes.

7.2.2 Learning 3D Representations from 2D Observations.

Latent 3D-aware representations. An area of 3D learning that studies learning from and generating 2D images has received a lot of attention in the last two years. Rhodin *et al.* (2018) train an auto-encoder that generates a view of a person (target) given a different view depicting the same person (source). Crucially, the latent representation is treated as if it were a 3D coordinate and a known 3D affine transformation that connects the source and the target views is applied to it. DeepVoxels by Sitzmann *et al.* (2019a) learn a voxel grid of features (instead of occupancies as was done previously) and apply 3D perspective transformations to the voxel grid (Yan *et al.*, 2016) before feeding them to a network that renders the target view. DeepVoxels are trained on a large collection of posed images of the same objects, with camera poses computed with traditional techniques for bundle adjustment (Triggs *et al.*, 1999). Scene Representation Networks (SRNs) by Sitzmann *et al.* (2019b) learn an implicit representation of a scene for novel view synthesis. When evaluated at a given 3D coordinate their implicit function predicts a feature vector (instead of a single occupancy value) which is subsequently decoded to obtain a pixel value. Coordinates of an intersection of the camera ray with the object surface are obtained by differentiable ray marching using a recurrent network (Hochreiter and Schmidhuber, 1997). SRNs extend the implicit function framework to represent appearance in addition to geometry and are trained end-to-end from a collection of posed images. Compared to our work, the aforementioned papers require to estimate camera pose for the images and are not able to do so end-to-end.

Nguyen-Phuoc *et al.* (2019) learn a latent 3D representation for view synthesis without requiring posed images, in an unsupervised manner. The method operates in a GAN framework where a discriminator is trained to classify real and generated views of 3D scenes. During training, they sample a latent vector z that encodes shape and appearance and a 3D rotation R, corresponding to the camera viewpoint. Conditioned on z, the generator network predicts a voxel grid of features and then applies a rotation R to this grid to align it with the camera axis. Finally, the decoder network renders the image given the aligned voxelized features. Owing to the GAN framework, HoloGAN is capable of learning 3D-aware representations without camera pose supervision and without requiring multiple posed views of the same object, making it suitable for few-shot learning. Mustikovela *et al.* (2020) build on HoloGAN to train a self-supervised view-point estimation network. BlockGAN (Nguyen-Phuoc *et al.*, 2020) extend HoloGAN to synthesise scenes composed of multiple objects and background in an unsupervised manner.

Differentiable rendering of meshes. Differentiable rendering is a more direct approach to learning 3D structures from 2D observations and received a lot of attention recently. Rendering aims to produce an image of a scene given the scene geometry, the appearance and the camera parameters. Differentiable rendering involves computing derivatives of the output image with respect to these latent variables. This allows to optimize for the underlying 3D shape by minimizing the loss between the generated and observed images and sometimes referred to as inverse graphics. The difficulty is that traditional rendering is a fundamentally discrete procedure: in order to compute a pixel value, the renderer must identify the triangles that intersect with the corresponding camera ray and select the one closest to the camera (eg. depth test based on the Z-buffer). OpenDR (Loper and Black, 2014) was one of the earlier differentiable renderers of polygon meshes that kept the standard rasterization pipeline for the forward process (based on OpenGL), while providing a way to compute approximate gradients. Liu *et al.* (2019a) propose a SoftRas renderer which is composed of intrinsically differentiable operations. Firstly, authors devise a mechanism for assigning a contribution of a triangle to a pixel intensity in a soft manner based on the distance to the boundary. Secondly, they propose a probabilistic Z-buffer that aggregates intensities based on the depth values. Where the traditional z-test selects the triangle with the minimum depth value, which is a discrete non-differentiable operation, Softras approximates the min function with softmin/softmax that is extensively used in Machine Learning. This allows to update positions of not only the visible triangles, but the occluded ones as well. Chen *et al.* (2019a) is a related work that proposes interpolation-based renderer and supports differentiation wrt. the appearance (texture) under Phong and Lambertian lighting models.

Differentiable rendering of implicit functions. The differentiable renderers described so far rely on deforming a template mesh and therefore have difficulties representing arbitrary topologies. Liu *et al.* (2019b) propose a framework to learn implicit fields with multi-view supervision. The main contribution is to sample implicit function at a set of anchor points selected with an importance sampling scheme with an emphasis on locations around the surface boundary. Niemeyer *et al.* (2020) propose Differentiable Volumetric Rendering (DVR) that can learn implicit representations from posed RGB images. Given a camera pose and a pixel coordinate, a ray is cast in the scene and the depth d is computed by probing the implicit function at fixed intervals until the occupancy prediction crosses the threshold from unoccupied to occupied. The depth d is then lifted to the 3D coordinate and is fed to an implicit texture function that computes an RGB value. The key contribution in DVR is an efficient analytic solution for the gradient of predicted depth using implicit differentiation. The method demonstrates excellent results on the real-world multi-view images competitive with the highly tuned classic approaches for 3D reconstruction. Concurrent work by Liu *et al.* (2020) proposes a differentiable renderer of signed distance functions using sphere tracing and supports various types of observations including depth images, surface normals and silhouettes. Jiang *et al.* (2020) present SDFDiff which is another approach for differentiable rendering of SDFs based on ray marching via sphere tracing and verify their design by experiments on multi-view and single-image 3D reconstruction.

Differentiable rendering of point clouds. Another group of recent work related to ours revolves around using point clouds as a representation for neural rendering of novel views. Meshry *et al.* (2019) run a full 3D reconstruction of the scene using classical Structure From Motion (SfM) and Multiview Stereo (MVS) algorithms and obtain a point cloud with registered images. For a given image the point cloud is projected into a deferred shading buffer containing colour, depth and semantic label which is then re-rendered by a neural network in order to reconstruct original image. The re-rendering network also takes as input a latent vector that encodes appearance allowing it to capture the distribution of scene appearances (time of day, season, weather etc.). Aliev *et al.* (2020) also utilize point clouds as a proxy representation for novel view synthesis. Each point is augmented with a learnable descriptor and the point cloud is then rasterized onto an image pyramid at various resolutions. Projecting points on a surface is usually accompanied by a *bleeding* problem, where points from the occluded surfaces can be seen through the holes. A go-to approach is to represent a point with a 3D disk, also known as splatting, while the multi-resolution projection of Aliev *et al.* (2020) deals with bleeding implicitly. Finally, a rendering U-Net transforms the rasterized points into realistic images. Wiles *et al.* (2020) propose a model for predicting novel views from a single image and is trained on pairs of registered views without 3D supervision. Given an input image the network predicts dense depth and per-pixel features which are projected onto a target view via a novel differentiable point cloud renderer.

Unsupervised learning of 3D. A recent method of Wu *et al.* (2020) trains an autoencoder that reconstructs depth, albedo, lighting and camera viewpoint from a single image without supervision for any of the quantities. Depth and albedo maps are predicted in a canonical orientation and are rendered (Kato *et al.*, 2018) with the lighting and the viewpoint to reconstruct the original image. The paper specifically aims at object categories exhibiting bilateral symmetry (such as faces) and adds an additional loss term on reconstructing the image from horizontally flipped depth and albedo maps. The reconstruction loss corresponds to a Laplace distribution and includes per-pixel uncertainty maps, which, in the case of the symmetric reconstruction, represent which parts of the image are not symmetric. The method requires only single views for training and is tested on large-scale face datasets demonstrating state-of-the-art 3D reconstruction while outperforming methods trained with 2D keypoint annotations.

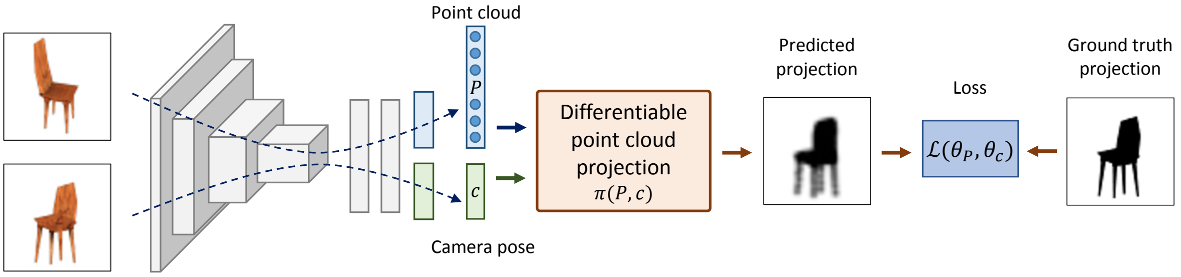

Point cloud
P
c
Camera pose
Differentiable point cloud projection $\pi(P, c)$
Predicted projection
Loss
$\mathcal{L}(\theta_P, \theta_c)$
Ground truth projection

Figure 7.1: Learning to predict the shape and the camera pose. Given two views of the same object, we predict the corresponding shape (represented as a point cloud) and the camera pose. Then we use a differentiable projection module to generate the view of the predicted shape from the predicted camera pose. Dissimilarity between this synthesized projection and the ground truth view serves as the training signal.

7.3 SINGLE-VIEW SHAPE AND POSE ESTIMATION

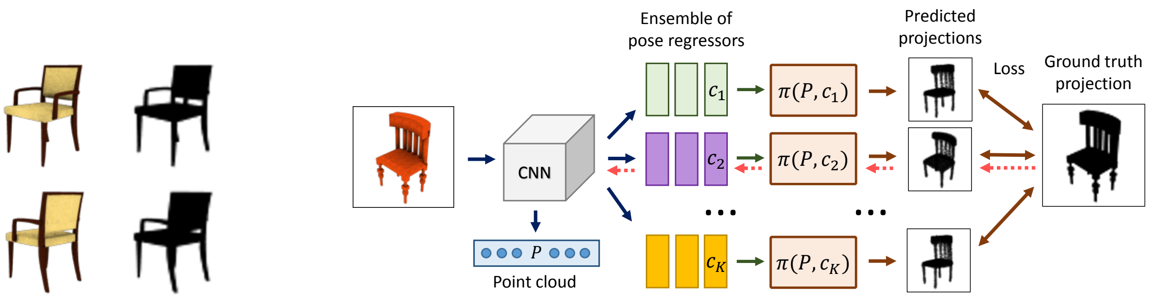
We address the task of predicting the three-dimensional shape of an object and the camera pose from a single view of the object. Assume we are given a dataset D of views of K objects, with m_i views available for the i-th object: $D = \cup_{i=1}^{K} \{ \left\langle \mathbf{x}_j^i, \mathbf{p}_j^i \right\rangle \}_{j=1}^{m_i}$. Here $\mathbf{x}_j^i$ denotes a color image and $\mathbf{p}_j^i$ – the projection of some modality (silhouette, depth map of a color image) from the same view. Each view may be accompanied with the corresponding camera pose c_j^i, but the more interesting case is when the camera poses are not known. We focus on this more difficult scenario in the remainder of this section.

An overview of the model is shown in Figure 7.1. Assume we are given two images $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same object. We use parametric function approximators to predict a 3D shape (represented by a point cloud) from one of them $\hat{P}_1 = F_P(\mathbf{x}_1, \theta_P)$, and the camera pose from the other one: $\hat{c}_2 = F_c(\mathbf{x}_2, \theta_c)$. In our case, F_P and F_c are convolutional networks that share most of their parameters. Both the shape and the pose are predicted as fixed-length vectors using fully connected layers.

Given the predictions, we render the predicted shape from the predicted view: $\hat{\mathbf{p}}_{1,2} = \pi(\hat{P}_1, \hat{c}_2)$, where π denotes the differentiable point cloud renderer described in Section 7.4. The loss function is then the discrepancy between this predicted projection and the ground truth. We use standard MSE in this work both for all modalities, summed over the whole dataset:

$$\mathcal{L}(\theta_P, \theta_c) = \sum_{i=1}^{N} \sum_{j_1, j_2 = 1}^{m_i} \left\| \hat{\mathbf{p}}_{j_1, j_2}^i - \mathbf{p}_{j_2}^i \right\|^2 . \tag{7.1}$$

Intuitively, this training procedure requires that for all pairs of views of the same object, the renderings of the predicted point cloud match the provided ground truth views.



(a) Pose ambiguity

(b) Training an ensemble of pose regressors

Figure 7.2: (a) Pose ambiguity: segmentation masks, which we use for supervision, look very similar from different camera views. (b) The proposed ensemble of pose regressors designed to resolve this ambiguity. The network predicts a diverse set $\{c_k\}_{k=1}^{K}$ of pose candidates, each of which is used to compute a projection of the predicted point cloud P. The weight update (backward pass shown in dashed red) is only performed for the pose candidate yielding the projection that best matches the ground truth.

Estimating pose with a distilled ensemble. We found that the basic implementation described above fails to predict accurate poses. This is caused by local minima: the pose predictor converges to either estimating all objects as viewed from the back, or all viewed from the front. Indeed, based on silhouettes, it is difficult to distinguish between certain views even for a human, see Figure 7.2 (a).

To alleviate this issue, instead of a single pose regressor $F_c(\cdot, \theta_c)$, we introduce an ensemble of K pose regressors $F_c^k(\cdot, \theta_c^k)$ (see Figure 7.2 (b)) and train the system with the "hindsight" loss (Guzmán-rivera *et al.*, 2012; Chen and Koltun, 2017):

$$\mathcal{L}_h(\theta_P, \theta_c^1, \ldots, \theta_c^K) = \min_{k \in [1,K]} \mathcal{L}(\theta_P, \theta_c^k). \tag{7.2}$$

The idea is that each of the predictors learns to specialize on a subset of poses and together they cover the whole range of possible values. No special measures are needed to ensure this specialization: it emerges naturally as a result of random weight initialization if the network architecture is appropriate. Namely, the different pose predictors need to have several (at least 3, in our experience) non-shared layers.

In parallel with training the ensemble, we distill it to a single regressor by using the best model from the ensemble as the teacher. This best model is selected based on the loss, as in Eq. (7.2). At test time we discard the ensemble and use the distilled regressor to estimate the camera pose. The loss for training the student is computed as an angular difference between two rotations represented by quaternions: $L(q_1, q_2) = 1 - \text{Re}(q_1 q_2^{-1} / \left\| q_1 q_2^{-1} \right\|)$, where Re denotes the real part of the quaternion. We found that standard MSE loss performs poorly when regressing rotation.

Network architecture. We implement the shape and pose predictor with a convolutional network with two branches. The network starts with a convolutional encoder with a total of 7 layers. The first one has a 5×5 kernel with 16 channels and stride 2. The remaining layers all have 3 kernels and come in pairs. The first layer in the pair has stride 2, the second one – stride 1. The number of channels grows by a factor of 2 after each strided layer. The convolutional encoder is followed by two fully connected layers with 1024 units. Then the network separates into two branches predicting shape and pose. The shape branch is an MLP with one hidden layer with 1024 units. The point cloud of N points is predicted as a vector with dimensionality $3N$ (point positions) or $6N$ (positions and RGB values).

The pose branch has one shared hidden layer with 1024 units. In the naive variant of the method, pose is predicted directly from this hidden layer. In the full approach with an ensemble of pose predictors, this layer is followed by 2 separate hidden layers for each pose predictor in the ensemble, with 32 units each. The camera pose is predicted as a quaternion. In the ensemble model we use $K = 4$ pose predictors. The "student" model is another branch with the same architecture. We used leaky ReLU with the negative slope 0.2 after all layers except for the shape prediction layer where we used the *tanh* non-linearity to constrain the output coordinates and for the pose prediction layer which is unrestrained.

7.4 DIFFERENTIABLE POINT CLOUDS

A key component of our model is the differentiable point cloud renderer π. Given a point cloud P and a camera pose c, it generates a view $\mathbf{p} = \pi(P, c)$. The point cloud may have a signal, such as color, associated with it, in which case the signal can be projected to the view.

The high-level idea of the method is to smooth the point cloud by representing the points with density functions. Formally, we assume the point cloud is a set of N tuples $P = \{\langle \mathbf{x}_i, \mathbf{s}_i, \mathbf{y}_i \rangle\}_{i=1}^{N}$, each including the point position $\mathbf{x}_i = (\cdot_{i,1}, \cdot_{i,2}, \cdot_{i,3})$, the size parameter $\mathbf{s}_i$, and the associated signal $\mathbf{y}_i$ (for instance, an RGB color). In most of our experiments the size parameter is a two-dimensional vector including the covariance of an isotropic Gaussian and a scaling factor. However, in general $\mathbf{s}_i$ can represent an arbitrary parametric distribution: for instance, in the supplement we show experiments with Gaussians with a full covariance matrix. The size parameters can be either specified manually or learned jointly with the point positions.

The overall differentiable rendering pipeline is illustrated in Figure 7.3. For illustration purposes we show 2D-to-1D projection in the figure, but in practice we perform 3D-to-2D projection. We start by transforming the positions of points to the standard coordinate frame by the projective transformation T_c corresponding to the camera pose c of interest: $\mathbf{x}'_i = T_c \mathbf{x}_i$. The transform T_c accounts for both extrinsic and intrinsic camera parameters. We also compute the transformed size parameters $\mathbf{s}'$ (the exact transformation rule depends on the distribution used). We set up the camera transformation matrix such that after the transform, the projection amounts to orthogonal projection along the third axis.

To allow for the gradient flow, we represent each point $\langle \mathbf{x}_i, \mathbf{s}_i \rangle$ by a smooth function $f_i(\cdot)$. In this work we set f_i to scaled Gaussian densities. The occupancy

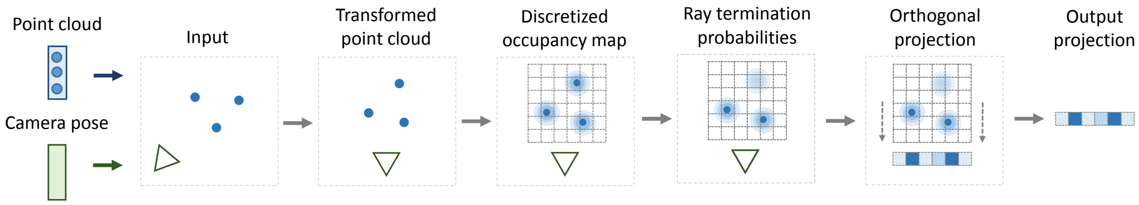


Figure 7.3: Differentiable rendering of a point cloud. We show 2D-to-1D projection for illustration purposes, but in practice we perform 3D-to-2D projection. The points are transformed according to the camera parameters, smoothed, and discretized. We perform occlusion reasoning via a form of ray tracing, and finally project the result orthogonally.

function of the point cloud is a clipped sum of the individual per-point functions:

$$o(\mathbf{x}) = \mathrm{clip}(\sum_{i=1}^{N} f_i(\mathbf{x}), [0,1]), \qquad f_i(\mathbf{x}) = c_i \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}'_i)^T \Sigma_i^{-1} (\mathbf{x}-\mathbf{x}'_i)\right), \quad (7.3)$$

where $\langle c_i, \Sigma_i \rangle = \mathbf{s}_i$ are the size parameters. We discretize the resulting function to a grid of resolution $D_1 \times D_2 \times D_3$. Note that the third index corresponds to the projection axis, with index 1 being the closest to the camera and D_3 – the furthest from the camera.

Before projecting the resulting volume to a plane, we need to ensure that the signal from the occluded points does not interfere with the foreground points. To this end, we perform occlusion reasoning using a differentiable ray tracing formulation, similar to the work of Tulsiani *et al.* (2017c). We convert the occupancies o to ray termination probabilities r as follows:

$$r_{k_1,k_2,k_3} = o_{k_1,k_2,k_3} \prod_{u=1}^{k_3-1} (1 - o_{k_1,k_2,u}) \;\; if \;\; k_3 \leqslant D_3, \qquad r_{k_1,k_2,D_3+1} = \prod_{u=1}^{D_3} (1 - o_{k_1,k_2,u}). \quad (7.4)$$

Intuitively, a cell has high termination probability r_{k_1,k_2,k_3} if its occupancy value o_{k_1,k_2,k_3} is high and all previous occupancy values $\{o_{k_1,k_2,u}\}_{u<k_3}$ are low. The additional background cell r_{k_1,k_2,D_3+1} serves to ensure that the termination probabilities sum to 1.

Finally, we project the volume to the plane:

$$p_{k_1,k_2} = \sum_{k_3=1}^{D_3+1} r_{k_1,k_2,k_3} y_{k_1,k_2,k_3}. \quad (7.5)$$

Here y is the signal being projected, which defines the modality of the result. To obtain a silhouette, we set $y_{k_1,k_2,k_3} = 1 - \delta_{k_3,D_3+1}$. For a depth map, we set $y_{k_1,k_2,k_3} = k_3/D_3$. Finally, to project a signal $\mathbf{y}$ associated with the point cloud, such as color, we set y to a discretized version of the normalized signal distribution: $\mathbf{y}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{y}_i f_i(\mathbf{x}) / \sum_{i=1}^{N} f_i(\mathbf{x})$.

7.4.1 Implementation Details

Technically, the most complex part of the algorithm is the conversion of a point cloud to a volume. We have experimented with two implementations of this step: one that is simple and flexible (we refer to it as basic) and another version that is less flexible, but much more efficient (we refer to it as fast). We implemented both versions using standard Tensorflow (Abadi *et al.*, 2016) operations. At a high level, in the basic implementation each function f_i is computed on an individual volumetric grid, and the results are summed. This allows for flexibility in the choice of the function class, but leads to both computational and memory requirements growing linearly with both the number of points N and the volume of the grid V, resulting in the complexity $O(NV)$. The fast version scales more gracefully, as $O(N+V)$. This comes at the cost of using the same kernel for all functions f_i. The fast implementation performs the operation in two steps: first putting all points on the grid with trilinear interpolation, then applying a convolution with the kernel.

Assume we are given a set of N points with coordinates and sizes $\{(\mathbf{x}_n, \sigma_n)\}_{n=0}^{N-1}$, as well as the desired spatial dimensions $D_1 \times D_2 \times D_3$ of the volume to be used for projection. Here we assume indexing of all tensors is 0-based.

In the basic implementation, we start by creating a coordinate tensor $\mathbf{M}$ of dimensions $N \times D_1 \times D_2 \times D_3 \times 3$ with entries $\mathbf{M}_{n,k_1,k_2,k_3,i} = k_i/D_i - 0.5$. Next, for each point we compute the corresponding Gaussian:

$$G_{n,k_1,k_2,k_3} = \exp(-0.5\sigma_n^{-2} \left\|\mathbf{M}_{n,k_1,k_2,k_3} - \mathbf{x}_n\right\|^2). \tag{7.6}$$

Finally, we sum these to get the resulting volume: $\mathbf{o}_{k_1,k_2,k_3} = \sum_{n=0}^{N-1} G_{n,k_1,k_2,k_3}$. This implementation is simple and allows for independently changing the sizes of points. However, on the downside, both memory and computation requirements scale linearly with the number of points.

Since linear scaling with the number of points makes large-scale experiments impractical, we implemented the fast version of the method that has lower computation and memory requirements. We implement the conversion procedure as a composition of trilinear interpolation and a convolution. Efficiency comes at the cost of using the same kernel for all points. We implemented trilinear interpolation using the Tensorflow scatter_nd function. We used standard 3D convolutions for the second step. For improved efficiency, we factorized them into three 1D convolutions along the three axes.

7.5 EXPERIMENTS

7.5.1 Experimental Setup

Datasets. We conduct the experiments on 3D models from the ShapeNet (Chang *et al.*, 2015) dataset. We focus on 3 categories typically used in related work: chairs, cars, and airplanes. We follow the train/test protocol and the data generation procedure of Tulsiani *et al.* (2017c): split the models into training, validation and test sets and render 5 random views of each model with random light source positions and random camera azimuth and elevation, sampled uniformly from $[0°, 360°)$ and $[-20°, 40°]$ respectively. To extract a point cloud from the ground truth meshes, we used the vertex densification procedure of Lin *et al.* (2018).

Evaluation metrics. We use the Chamfer distance as our main evaluation metric, since it has been shown to be well correlated with human judgment of shape similarity (Sun *et al.*, 2018). Given a ground truth point cloud $P^{gt} = \{\mathbf{x}_n^{gt}\}$ and a predicted point cloud $P^{pr} = \{\mathbf{x}_n^{pr}\}$, the distance is defined as follows:

$$d_{Chamf}(P^{gt}, P^{pred}) = \frac{1}{|P^{pr}|} \sum_{\mathbf{x}^{pr} \in P^{pr}} \min_{\mathbf{x} \in P^{gt}} \|\mathbf{x}^{pr} - \mathbf{x}\|_2 + \frac{1}{|P^{gt}|} \sum_{\mathbf{x}^{gt} \in P^{gt}} \min_{\mathbf{x} \in P^{pr}} \left\|\mathbf{x}^{gt} - \mathbf{x}\right\|_2 . \tag{7.7}$$

The two sums in Eq. (7.7) have clear intuitive meanings. The first sum evaluates the precision of the predicted point cloud by computing how far on average is the closest ground truth point from a predicted point. The second sum measures the coverage of the ground truth by the predicted point cloud: how far is on average the closest predicted point from a ground truth point.

For measuring the pose error, we use the same metrics as Tulsiani *et al.* (2018): accuracy (the percentage of samples for which the predicted pose is within 30° of the ground truth) and the median error (in degrees). We computed the angular difference between two rotations represented with quaternions q_1 and q_2 as $2\,\mathrm{acos}\left(q_1 q_2^{-1} / \left\|q_1 q_2^{-1}\right\|\right)$. Before starting the pose and shape evaluation, we align the canonical pose learned by the network with the canonical pose in the dataset, using Iterative Closest Point (ICP) algorithm on the first 20 models in the validation set.

For the outputs of voxel-based methods, we extract the surface mesh with the marching cubes algorithm and sample roughly 10000 points from the computed surface. We tuned the threshold parameters of the marching cubes algorithm based on the Chamfer distance on the validation set.

Training details. We trained the networks using the Adam optimizer (Kingma and Ba, 2015), for 600@000 mini-batch iterations. We used mini-batches of 16 samples (4 views of 4 objects). We used a fixed learning rate of 0.0001 and the standard momentum parameters. We used the fast projection in most experiments, unless mentioned otherwise. We varied both the number of points in the point cloud and the resolution of the volume used in the projection operation depending on the resolution of the ground truth projections used for supervision. We used the volume with the same side as the training samples (e.g., 64^3 volume for 64^2 projections), and we used 2000 points for 32^2 projections, 8000 points for 64^2 projections, and 16@000 points for 128^2 projections.

When predicting dense point clouds, we have found it useful to apply dropout to the predictions of the network to ensure even distribution of points on the shape. Dropout effects in selecting only a subset of all predicted points for projection and loss computation. In experiments reported in Sections 7.5.3 and 7.5.5 we started with a very high 90% dropout and linearly reduced it to 0 towards the end of training. We

	Full	No drop.	Fixed $\sigma = 1.6$	Fixed scale	4000 pts.	2000 pts.	1000 pts.
Precision	2.05	2.60	2.06	2.01	2.10	2.17	2.11
Coverage	1.98	1.99	2.82	2.26	2.19	2.54	2.98
Chamfer	4.03	4.59	4.89	4.27	4.28	4.70	5.10

Table 7.1: Ablation study of our method for shape prediction. We report the Chamfer distance between normalized point clouds, multiplied by 100, as well as precision and coverage.

also implemented a schedule for the point size parameters, linearly decreasing from 5% of the projection volume size to 0.3% over the course of training. The scaling coefficient of the points was learned in all experiments.

Computational efficiency. A practical advantage of a point-cloud-based method is that it does not require using a 3D convolutional decoder as required by voxel-based methods. This improves the efficiency and allows the method to better scale to higher resolution. For resolution 32 the training times of the methods are roughly on par. For 64 the training time of our method is roughly 1 day in contrast with 2.5 days for its voxel-based counterpart. For 128 the training time of our method is 3 days, while the voxel-based method does not fit into 12Gb of GPU memory with our batch size.

7.5.2 Ablation study

We evaluate the effect of different components of the model on the shape prediction quality. We measure these by training with pose supervision on ShapeNet chairs, with 64^2 resolution of the training images. Results are presented in Table 7.1. The "Full" method is trained with 8000 point, point dropout, sigma schedule, and learned point scale. All our techniques are useful, but generally the method is not too sensitive to these.

7.5.3 Estimating Shape with Known Pose

Comparison with baselines. We start by benchmarking the proposed formulation against existing methods in the simple setup with known ground truth camera poses and silhouette-based training. We compare to Perspective Transformer Networks (PTN) of Yan *et al.* (2016), Differentiable Ray Consistency (DRC) of Tulsiani *et al.* (2017c), Efficient Point Cloud Generation (EPCG) of Lin *et al.* (2018), and to the voxel-based counterpart of our method. PTN and DRC are only available for 32^3 output voxel grid resolution. EPCG uses the point cloud representation, same as our method. However, in the original work EPCG has only been evaluated in the unrealistic setup of having 100 random views per object and pre-training from 8 fixed views (corners of a cube). We re-train this method in the more realistic setting

	Resolution 32				Resolution 64		Resolution 128	
	DRC	PTN	Ours-V	Ours	Ours-V	Ours	EPCG	Ours
Airplane	8.35	3.79	5.57	4.52	4.94	3.50	4.03	**2.84**
Car	4.35	3.94	3.88	4.22	3.41	2.98	3.69	**2.42**
Chair	8.01	5.10	5.57	5.10	4.80	4.15	5.62	**3.62**
Mean	6.90	4.27	5.01	4.61	4.39	3.55	4.45	**2.96**

Table 7.2: Quantitative results on shape prediction with known camera pose. We report the Chamfer distance between normalized point clouds, multiplied by 100. Our point-cloud-based method (Ours) outperforms its voxel-based counterpart (Ours-V) and benefits from higher resolution training samples. Finally, we compare our results to the methods DRC (Tulsiani *et al.*, 2017c), PTN (Yan *et al.*, 2016) and EPCG (Lin *et al.*, 2018).

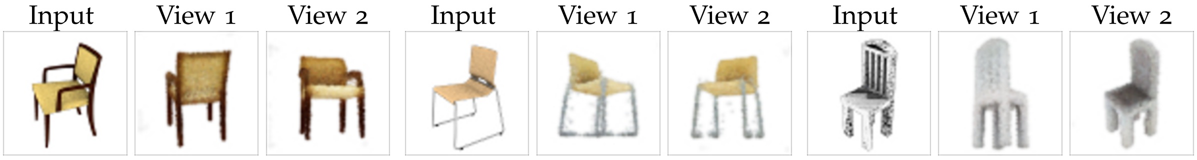


Figure 7.4: Learning colored point clouds. Best viewed on screen. We show the input image, as well as two renderings of the predicted point cloud from other views. The general color is preserved well, but the fine details may be lost.

used in this work – 5 random views per object.

The quantitative results are shown in Table 7.2. Our point-cloud-based formulation (Ours) outperforms its voxel-based counterpart (Ours-V) in all cases. It improves when provided with high resolution training signal, and benefits from it more than the voxel-based method. Overall, our best model (at 128 resolution) decreases the mean error by 30% compared to the best baseline. An interesting observation is that at low resolution, PTN performs remarkably well, closely followed by our point-cloud-based formulation. Note, however, that the PTN formulation only applies to learning from silhouettes and cannot be easily generalized to other modalities.

Our model achieves 50% improvement over the point cloud method EPCG, despite it being trained from depth maps, which is a stronger supervision compared to silhouettes used for our models. When trained with silhouette supervision only, EPCG achieves an average error of 8.20, 2.7 times worse than our model. We believe our model is more successful because our rendering procedure is differentiable w.r.t. all three coordinates of points, while the method of Lin et al. – only w.r.t. the depth.

Colored point clouds. Our formulation supports training with other supervision than silhouettes, for instance, color. In Figure 7.4 we demonstrate qualitative results of learning colored point clouds with our method. Despite challenges presented by

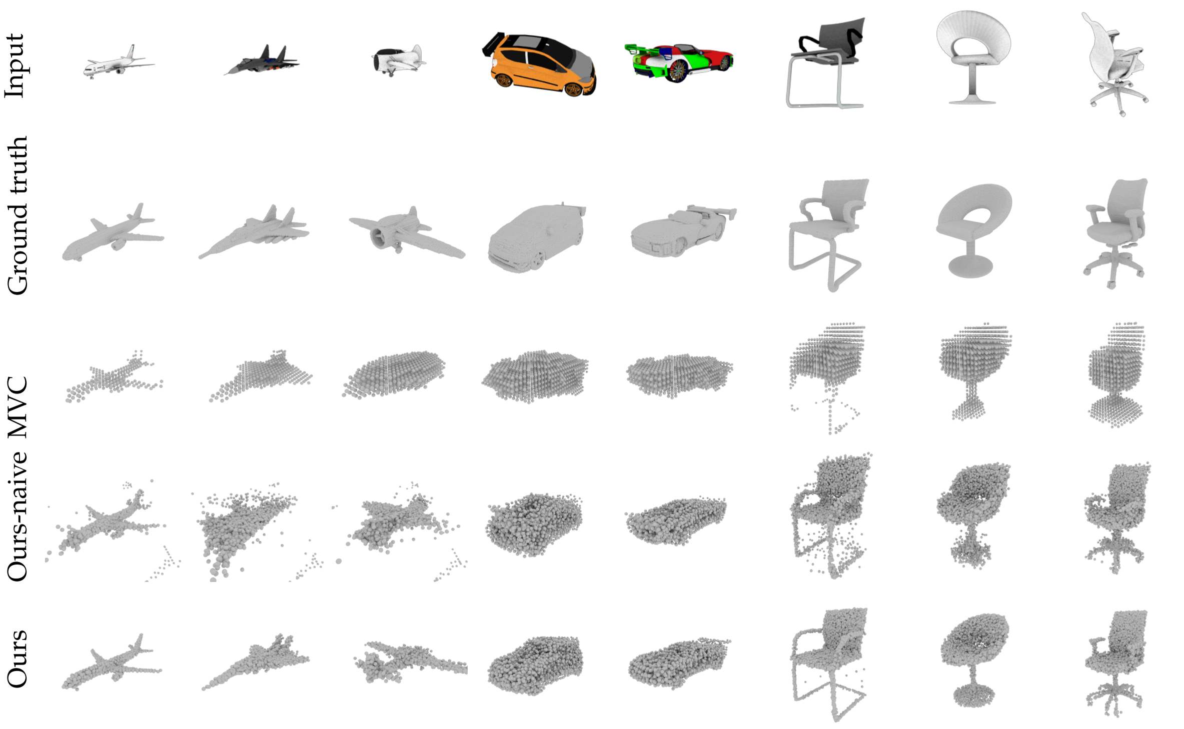


Figure 7.5: Qualitative results of shape prediction. Best viewed on screen. Shapes predicted by our naive model with a single pose predictor (Ours-naive) are more detailed than those of MVC (Tulsiani *et al.*, 2018). The model with an ensemble of pose predictors (Ours) generates yet sharper shapes. The point cloud representation allows to preserve fine details such as thin chair legs.

the variation in lighting and shading between different views, the method is able to learn correctly colored point clouds. For objects with complex textures the predicted colors get blurred (last example).

7.5.4 Towards Part-based Models

In the experiments reported so far the shape parameters of the points were set by hand, and only the scaling factor was learned. However, our formulation allows learning the shape parameters jointly with the positions of the points. Here we explore this direction using the basic implementation, since it allows for learning a separate shape for each point in the point set. We explore two possibilities: isotropic Gaussians, parametrized by a single scalar and general covariance matrices, parametrized by 7 numbers: 3 diagonal values and a quaternion representing the rotation (this is an overcomplete representation). This resembles part-based models: now instead of composing the object of "atomic" points, a whole object part can be represented by a single Gaussian of appropriate shape (for instance, an elongated Gaussian can represent a leg of a chair).

Figure 7.6 qualitatively demonstrates the advantage of the more flexible model

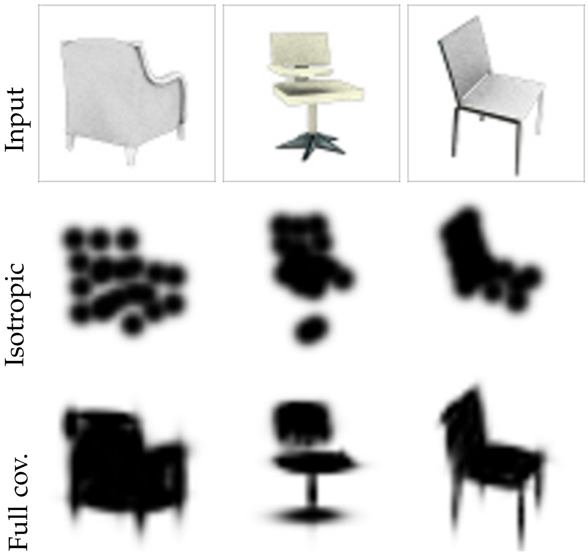


Figure 7.6: Silhouettes learned with full learned Gaussian covariance versus hand-tuned isotropic Gaussian, using 20 points.

over the simpler alternative with isotropic Gaussians. One could imagine employing yet more general and flexible per-point shape models, and we see this as an exciting direction of future work.

Figure 7.7 shows the projection error of different approaches for varying number of points in the set. Learnable parameters perform better than hand-tuned and learned full covariance performs better than learned isotropic covariance. A caveat is that training with full covariance matrix is computationally more heavy in our implementation.

7.5.5 Estimating Shape and Pose

We now drop the unrealistic assumption of having the ground truth camera pose during training and experiment with predicting both the shape and the camera pose. We use the ground truth at 64 pixel resolution for our method in these experiments. We compare to the concurrent Multi-View Consistency (MVC) approach of Tulsiani *et al.* (2018), using results reported by the authors for pose estimation and pre-trained models provided by the authors for shape evaluations.

Quantitative results are provided in Table 7.3. Our naive model (Ours-naive) learns quite accurate shape (7% worse than MVC), despite not being able to predict the pose well. Our explanation is that predicting wrong pose for similarly looking projections does not significantly hamper the training of the shape predictor. Shape predicted by the full model (Ours) is yet more precise: 28% more accurate than MVC and only 10% less accurate than with ground truth pose (as reported in Table 7.2). Pose prediction improves dramatically, thanks to the diverse ensemble formulation. As a result, our pose prediction results are on average slightly better than those of

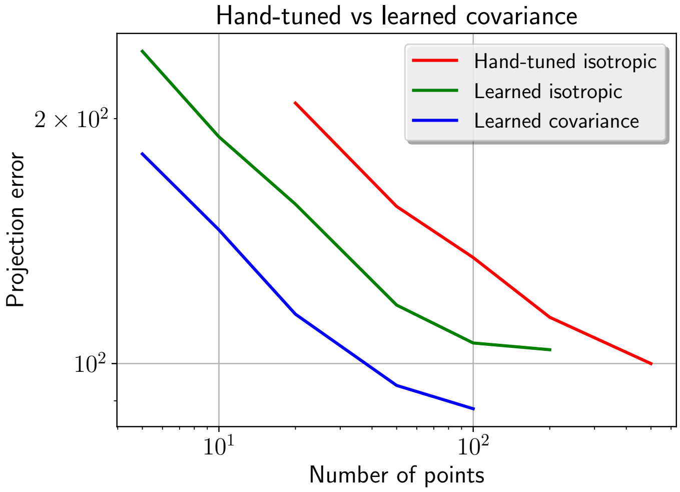


Figure 7.7: Projection error with different models and different number of points. More flexible density distributions allow for reaching the same error with fewer points. In particular, full learnable covariance can require roughly an order of magnitude fewer points than hand-tuned isotropic covariance to reach the same quality.

MVC (Tulsiani *et al.*, 2018) in both metrics, and even better in median error than the results of training with ground truth pose labels (as reported by Tulsiani *et al.* (2018)).

Figure 7.5 shows a qualitative comparison of shapes generated with different methods. Even the results of the naive model (Ours-naive) compare favorably to MVC (Tulsiani *et al.*, 2018). Introducing the pose ensemble leads to learning more accurate pose and, as a consequence, more precise shapes. These results demonstrate the advantage of the point cloud representation over the voxel-based one. Point clouds are especially suitable for representing fine details, such as thin legs of the chairs. We also show typical failure cases of the proposed method. One of the airplanes is rotated by 180 degrees, since the network does not have a way to find which orientation is considered correct. The shapes of two of the chairs somewhat differ from the true shapes. This is because of the complexity of the training problem and, possibly, overfitting. Yet, the shapes look detailed and realistic. Additional qualitative results are shown in Figure 7.8. Note that for MVC we use the binarization threshold that led to the best quantitative results.

7.5.6 Discovery of Semantic Correspondences

Besides higher shape fidelity, the "matter-centric" point cloud representation has another advantage over the "space-centric" voxel representation: there is a natural correspondence between points in different predicted point clouds. Since we predict points with a fully connected layer, the points generated by the same output unit in

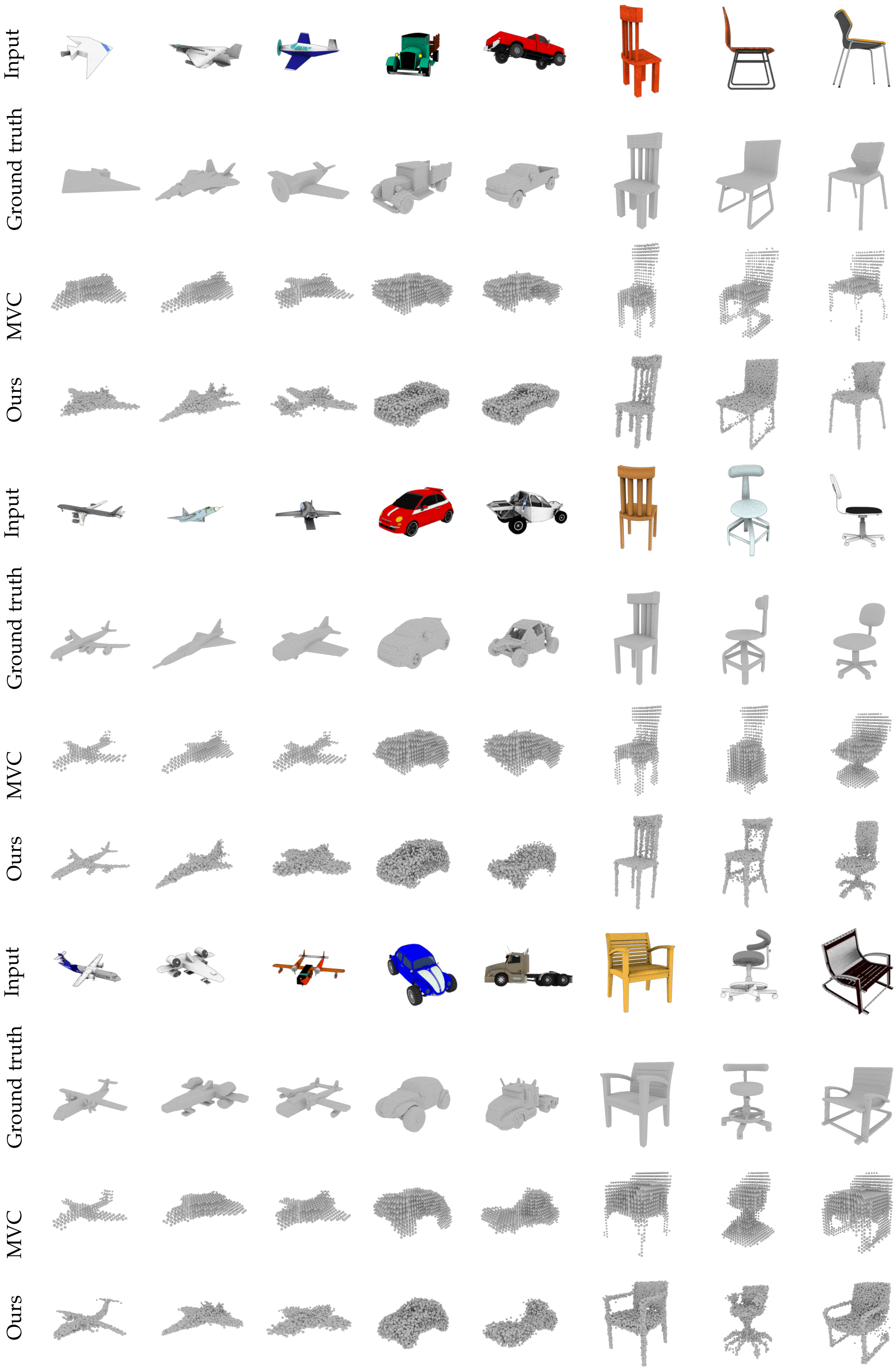


Figure 7.8: Additional qualitative results and comparisons with the method MVC (Tulsiani *et al.*, 2018).

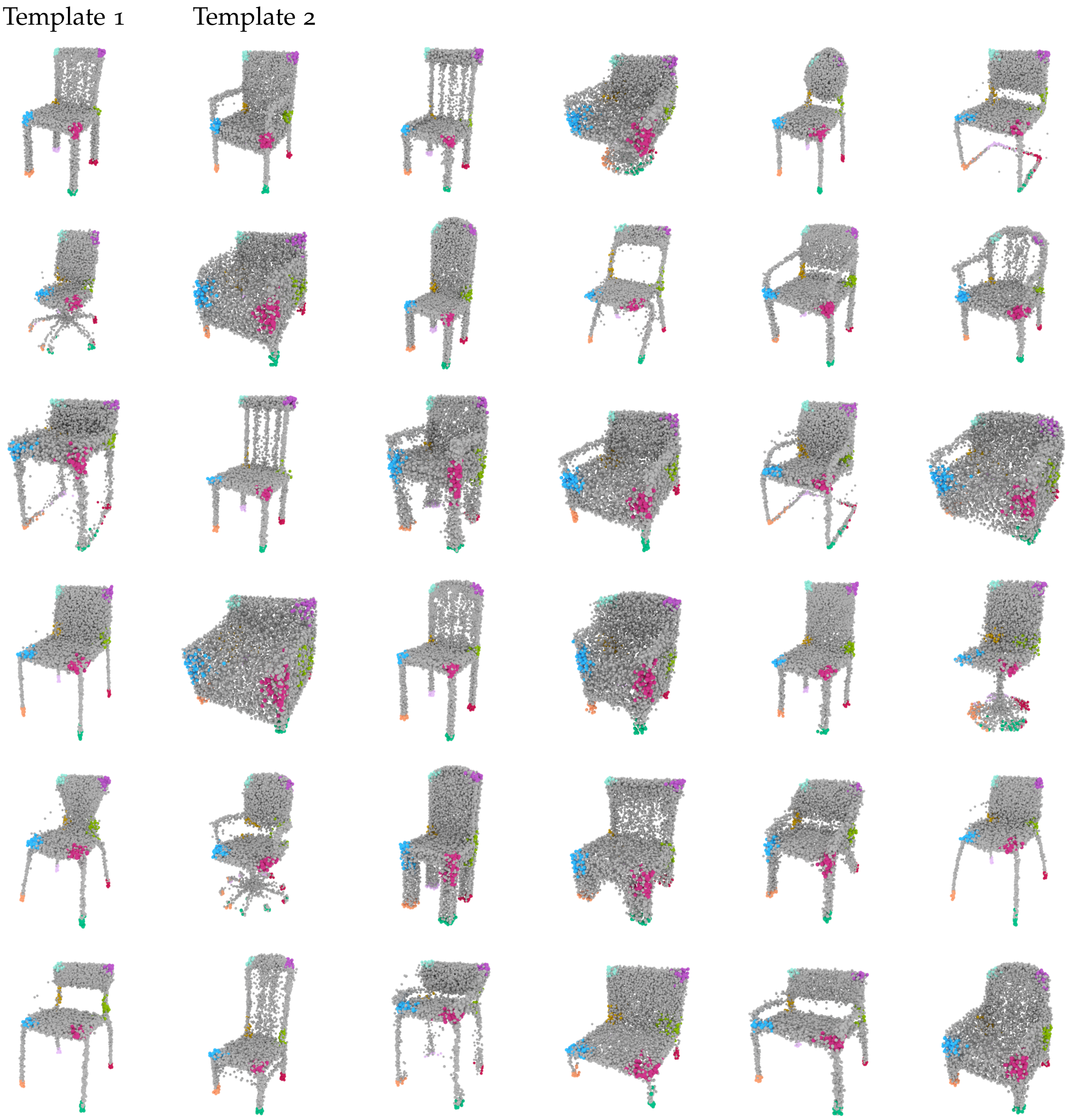


Figure 7.9: Discovered semantic correspondences. Points of the same color correspond to the same subset in the point cloud across different instances. The points were selected on two template instances (top left). Best viewed on screen.

	Shape (D_{Chamf})			Pose (Accuracy & Median error)							
	MVC	Ours-naive	Ours	GT pose		MVC		Ours-naive		Ours	
Airplane	4.43	7.22	**3.91**	**0.79**	10.7	0.69	14.3	0.20	100.2	0.75	**8.2**
Car	4.16	4.14	**3.47**	**0.90**	7.4	0.87	5.2	0.49	42.8	0.86	**5.0**
Chair	6.51	4.79	**4.30**	0.85	11.2	0.81	**7.8**	0.50	31.3	**0.86**	8.1
Mean	5.04	5.38	**3.89**	**0.85**	10.0	0.79	9.0	0.40	58.1	0.82	**7.1**

Table 7.3: Quantitative results of shape and pose prediction. Best results for each metric are highlighted in bold. The naive version of our method predicts the shape quite well, but fails to predict accurate pose. The full version predicts both shape and pose well. MVC and GT pose results are from the article of Tulsiani *et al.* (2018).

different shapes can be expected to carry similar semantic meaning. We empirically verify this hypothesis. We choose two instances from the validation set of the chair category as templates (shown in the top-left corner of Figure 7.9) and manually annotate 3D keypoint locations corresponding to characteristic parts, such as corners of the seat, tips of the legs, etc. Then, for each keypoint we select all points in the predicted clouds within a small distance from the keypoint and compute the intersection of the points indices between the two templates. (Intersection of indices between two object instances is not strictly necessary, but we found it to slightly improve the quality of the resulting correspondences.) We then visualize points with these indices on several other object instances, highlighting each set of points with a different color. Results are shown in Figure 7.9. As hypothesized, selected points tend to represent the same object parts in different object instances. Note that no explicit supervision was imposed towards this goal: semantic correspondences emerge automatically. We attribute this to the implicit ability of the model to learn a regular, smooth representation of the output shape space, which is facilitated by reusing the same points for the same object parts.

7.5.7 Interpolation of Shapes in the Latent Space

Fig. 7.10 shows results of linear interpolation between shapes in the latent space given by the first (shared) fully connected layer. We can observe gradual transitions between shapes, which indicates that the model learns a smooth representation of the shape space. Failure cases, such as legs in the second row, can be attributed to the limited representation of the office chairs with 5 legs in the dataset.

7.6 CONCLUSION

We have proposed a method for learning pose and shape of 3D objects given only their 2D projections, using the point cloud representation. Extensive validation has

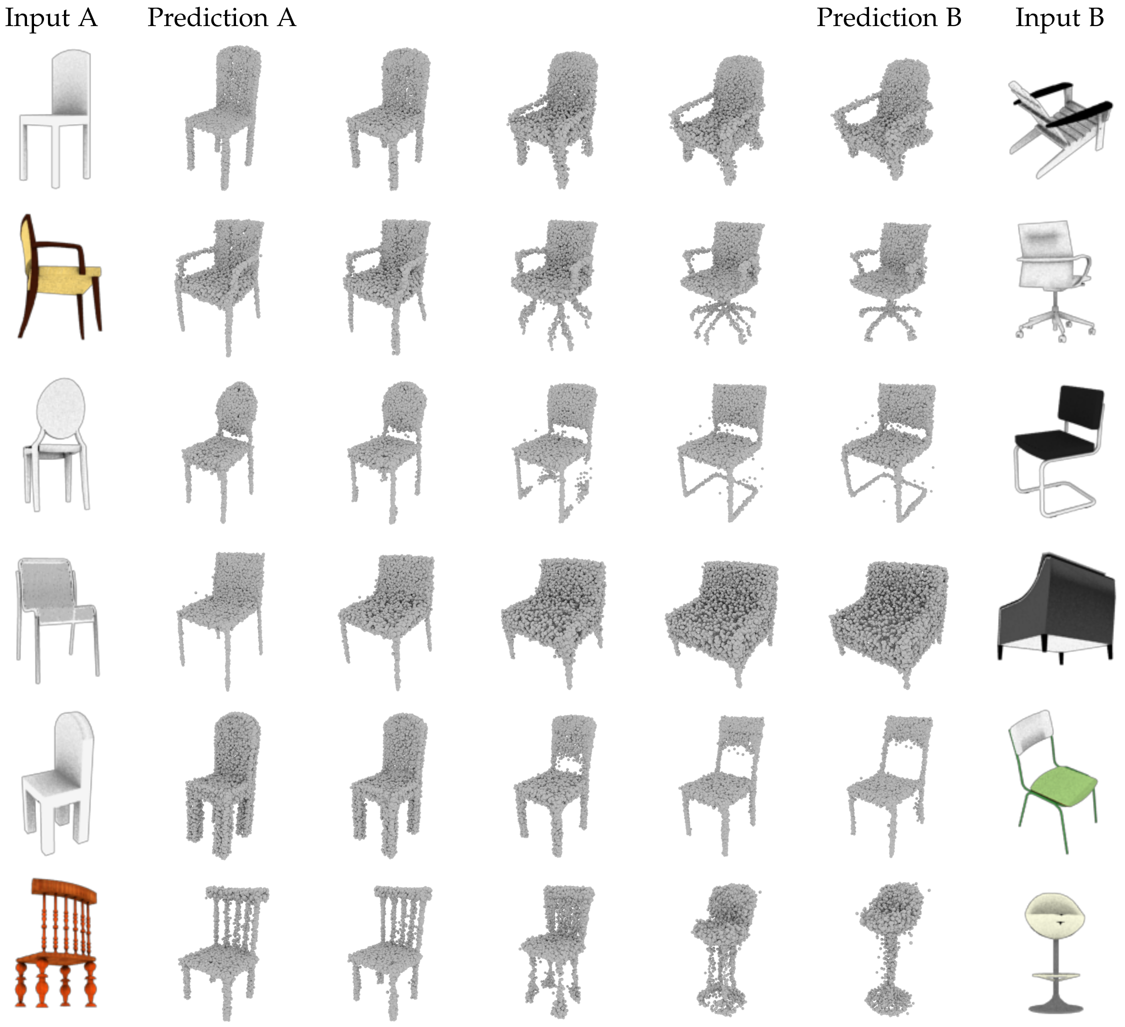


Figure 7.10: Interpolation of shapes in the latent space.

shown that point clouds compare favorably with the voxel-based representation in terms of efficiency and accuracy. Our work opens up multiple avenues for future research. First, our projection method requires an explicit volume to perform occlusion reasoning. We believe this is just an implementation detail, which might be relaxed in the future with a custom rendering procedure. Second, since the method does not require accurate ground truth camera poses, it could be applied to learning from real-world data. Learning from color images or videos would be especially exciting, but it would require explicit reasoning about lighting and shading, as well as dealing with the background. Third, we used a very basic decoder architecture for generating point clouds, and we believe more advanced architectures (Yang *et al.*, 2018) could improve both the efficiency and the accuracy of the method. Finally, the fact that the loss is explicitly computed on projections (in contrast with, e.g., work by Tulsiani *et al.* (2017c)), allows directly applying advanced techniques from the 2D domain, such as perceptual losses and GANs, to learning 3D representations.

8 CONCLUSIONS

In this chapter we summarise the main contributions of the thesis as well as outline potential avenues for future research that arise from this work.

8.1 CONTRIBUTIONS AND IMPACT

Multi-Person Pose Estimation. In Chapter 3 we introduced an approach for multi-person 2D human pose estimation in unconstrained real-world scenarios. It was built on three major advancements: 1) a strong keypoint detector based on ResNets 2) image conditioned-pairwise terms for efficient grouping of part-hypotheses into instances 3) hierarchical optimization scheme for graph partitioning to further reduce the runtime and improve performance. Our approach addressed multi-person pose estimation in a holistic manner and for the first time enabled highly accurate pose estimation for in-the-wild scenes. Together with our prior work (Pishchulin *et al.*, 2016) this opened up a fruitful avenue for research resulting in a significant amount of publications on multi-person pose estimation presented each year at major computer vision conferences.

Our multi-task ConvNet detects body joints and estimates pairwise relations between them in a single forward pass which allows integration of our system into real-time applications. This is in contrast to the top-down approaches which first estimate instance bounding boxes and then run a pose estimation network on the crops, with the run-time scaling linearly with the number of subjects in the scene. We presented a real-time demo of our approach at the ECCV 2016 conference and received largely positive feedback. Since the publication of our research there appeared a number interesting bottom-up pose estimation methods (please refer to Section 2.2 for a review). In parallel to our efforts bottom-up methods were also adopted in the object detection community (Redmon *et al.*, 2016; Liu *et al.*, 2016b; Lin *et al.*, 2017). Often referred to as single-shot detectors they offer significantly improved run-time compared to the top-down approaches such as Faster R-CNN Ren *et al.* (2015). Very similar to our work two recent papers presented bottom-up object detection methods based on grouping of extreme keypoints (Law and Deng, 2018; Zhou *et al.*, 2019).

We open sourced the implementation of the method so that the community could benefit from our research [1]. We are aware of at least three publications directly using our pose estimation code (Bogo *et al.*, 2016; Lassner *et al.*, 2017; Iqbal *et al.*, 2017b). Though the initial implementation of our method was done in Caffe framework

[1] https://github.com/eldar/deepcut and https://github.com/eldar/deepcut-cnn.

(Jia *et al.*, 2014), we later undertook an effort to create a Tensorflow port (Abadi *et al.*, 2016). A lot of attention has been given to the careful implementation with the goal of it being clean and modular. The corresponding GitHub repository [2] has reached almost 1000 stars which indicates of the significant interest to our work. Noteably, our algorithm served as a foundation of the open source toolkit DeepLabCut (deriving its name from our method) that integrates graphical user interface for rapid data annotation and our Tensorflow keypoint detector. DeepLabCut targets the neuroscience and biological communities and enables them to perform quantified analysis of motor behavior of animals. Since the publication of the original report (Mathis *et al.*, 2018) the framework has seen active development and received widespread adoption by hundreds of labs all over the world. The part detector algorithm described in Section 3.2.2 proved generic enough and generalised beyond humans to detection of landmarks of the wide range of animals including mammals, fish, insects, etc.

Articulated Multi-Person Tracking. In Chapter 4 we took a step further from multi-person pose estimation and introduced a novel problem of articulated multi-person tracking in videos. Articulated multi-person tracking unifies multi-target tracking Milan *et al.* (2016) and multi-person pose estimation Pishchulin *et al.* (2016). Our proposed solution to this problem based on the graph partitioning formulation which extends the framework in Chapter 3. It performs tracking and pose estimation in a holistic manner by optimising a single objective. They key contributions include a simplified formulation that allows real-time applications and an end-to-end objective for grouping of body parts. Our experiments demonstrated that such a holistic approach is able to exploit the complimentary information available in temporal domain and boost pose estimation accuracy over a per-frame baseline.

We are very delighted that the community showed interest in the problem of articulated multi-person tracking and we see new publications featuring frequently at major computer vision conferences (see Section 2.3 for an overview of the recent work). Our work and the works that followed immediately after largely relied on per-frame detection and temporal grouping and thus did not truly learn temporal dynamics. However, more recent work (Wang *et al.*, 2020; Snower *et al.*, 2020) demonstrates the benefits of end-to-end multi-person tracking and we expect even more research happening in this direction.

A dataset for articulated multi-person tracking. In Chapter 5 we introduced a large-scale dataset and a benchmark for articulated multi-person tracking. It is the first dataset of video sequences comprising complex multi-person scenes with fully annotated person tracks and 2D keypoints in each frame. We also provided evaluation metrics for multi-person pose estimation (Pishchulin *et al.*, 2016) and joint tracking accuracy (Milan *et al.*, 2016). We withheld the test set to ensure a fair comparison between different models and created a server that performs automatic evaluation of submissions. We organised two workshops at ICCV 2017 and ECCV 2018 alongside with the competitions in order to promote the dataset. The dataset was received well by the community and has become standard for the task of

[2] https://github.com/eldar/pose-tensorflow.

multi-person pose tracking with over 500 registered users, weekly submissions to our evaluation server and regular publications using the dataset. We hope that availability of such a large scale dataset would enable progress towards truly end-to-end pose estimation and tracking in complex real-world scenes.

Egocentric Marker-less 3D Motion Capture. In Chapter 6 we took a different approach to address human pose estimation in crowded scenes as compared to the traditional pose estimation paradigm covered in the previous chapters. While traditional methods require multi-view camera setups for robust 3D pose estimation and motion capture, we opted for on-body wearable cameras instead. Specifically, a pair of downward facing, head-mounted cameras and captures the entire human body. We combine a generative 3D body pose model with discriminatively trained CNN-based 2D body joint detectors operating on distorted fish-eye images. The CNN-based keypoint detector is trained on a novel dataset collected in a mocap studio with joints back-projected from 3D space using fish-eye camera model. Our system is able to perform motion capture in highly crowded scenarios in a variety of scenes without the constraints imposed by systems with external cameras. Since the publication of our work many commercial products such as the VR headset from Facebook utilise similar approaches for 3D pose estimation, in particular for hand pose estimation, which normally suffers from occlusion and lower resolution in traditional systems with external cameras. And with the advent of highly interactive applications in VR, such as highly immersive games, pose estimation using on-body sensors will continue to gain momentum.

Learning 3D Object Shape and Camera Pose. In Chapter 7 we introduce a method for estimating 3D shape of rigid objects from a single image with weak supervision in the form of 2D projections. We represent 3D shape as point clouds and propose a differentiable mechanism of projecting point clouds onto 2D plane. We also devise a mechanism for dealing with ambiguity of camera poses that, in contrast to prior approaches, allows to train our system without camera pose supervision. Our method takes its spot among the proliferation of work on end-to-end learning of 3D shape representations that appeared in the recent years. In particular, implicit representations had gained popularity in the research community due to their representational power, flexibility and efficiency. Nevertheless, point clouds still remain an attractive representation as they are the output of the classical 3D reconstruction algorithms and recent works

8.2 FUTURE WORK

We envision several possible directions for future research that follow up and further develop the work carried out in this thesis.

1) When we carried out the work of this thesis, the primary concern was always to achieve the best accuracy on established benchmarks with the considerations about run-time efficiency not being the primary focus. However, with the ubiquity of mobile devices there is a steady demand for re-distributing computation from

the cloud-based servers onto the devices themselves. Pose estimation and tracking algorithms that can perform comparatively well under the tight computational budget will be impactful going forward. We believe that taking advantage of the videos by exploiting inter-frame redundancy can bring significant gains for run-time efficiency. Indeed, the content changes very little between frames and very recent work (Bertasius *et al.*, 2019) demonstrates a possibility of light-weight propagation of poses from one frame to another in contrast to naively running a detector in each frame. Modeling global motion (eg. changing of camera position) and local motion (change in body pose) independently could provide further gains in efficiency. Incorporating some notion of smoothness in pose tracking can additionally simplify the design of models. Another promising direction for improving efficiency is performing adaptive computation in the spatial domain. More specifically, pose estimation coould benefit from allocating the computational budget on the regions of the image containing humans and ignoring the background. This is in contrast with the naive application of CNNs where each spatial location is processed by the same computational graph. The recent research (Verelst and Tuytelaars, 2020) demonstrates very promising results in this direction and we expect even more interesting work to follow.

2) Another possibility for improving both computational efficiency and accuracy is to utilise novel sensor modalities. In the past human pose estimation had already made significant leaps by utilising Kinect depth sensor (Shotton *et al.*, 2011a). We envision that going forward event-based sensors will provide further improvements. Event cameras are biologically inspired sensors that record brightness changes and offer many advantages over traditional RGB cameras including high temporal resolution, low latency and high dynamic range (Gallego *et al.*, 2019). Event cameras record a very sparse signal and operate at much lower power which makes them suitable for mobile and robotics applications. Recent works take advantage of high frame rates of event cameras and demonstrate real-time systems for gesture recognition (Amir *et al.*, 2017) and human motion capture (Xu *et al.*, 2020). We believe that these new sensor modalities can be complimentary to the traditional ones and new approaches that perform sensor fusion at the model level would yield even better results.

3) 2D pose estimation concerns itself with localising body joint locations in the image plane. However, knowing that the 2D pose is a projection of a 3D skeleton could provide additional information for training pose estimation models. Using joint velocity constraints and the extent of the articulation for supervision of the models could improve articulated tracking performance. Recent research is already exploring similar ideas in the context of multi-view geometry by obtaining cheap supervision (Simon *et al.*, 2017; Iskakov *et al.*, 2019) and we believe that the knowledge of the 3D structure of human anatomy and biomechanics can help improve pose estimation in a single-view setting too.

4) DeepLabCut (Mathis *et al.*, 2018) kick started widely accessible animal pose estimation which facilitates quantified behavior behavior analysis in the fields of biology in neuroscience. This enables research on how the nervous system drives

complex articulated movements that animals are capable of performing. Animal pose estimation presents unique challenges that go beyond the category of humans. The first one concerns itself with the immense diversity of the biological kingdom of animals with the researchers using pose estimation to study worms, arthropods, fish, reptiles and mammals among others. Each of these biological classes is itself comprised of wildly varying skeleton configurations and appearances in a multitude of species. This demands general solutions that do not rely on excessive data annotation and will allow researchers to rapidly set up pose estimation for the new species under investigation. Solving this challenge will require using different types of self-supervision. The second challenge worth noting is the multi-animal pose estimation setting. Imagine an anthill swarming with ants with the task of articulated multi-instance tracking. In this setting one cannot rely on appearance alone to distinguish individual instances and instead must truly track each instance. Such a problem is further exacerbated by the extreme crowdedness and a successful algorithm must be able to deal with partial visibility as well as temporary loss of sight of the tracked instances. Furthermore, this extreme example demands that methods scale well with the number of instances in the scene. Finally, given the diversity of animals it is also important to establish appropriate benchmarks. Any single species would not be sufficient to test an algorithm and collecting representative datasets would be too costly.

Animal pose estimation is now gaining momentum and more researchers are dedicating their efforts towards the problem. Recent work (Günel *et al.*, 2019) introduces an approach named DeepFly3D that applies multi-view bootstrapping and self-supervision for 3D pose estimation of adult Drosophila. LEAP (Pereira *et al.*, 2019) is a framework similar to DeepLabCut and provides a graphical user interface for rapid data annotation as well as fully convolutional part detector. The recent SMALST method (Zuffi *et al.*, 2019) performs 3D pose, shape and texture capture of animals from in the wild images using a SMAL animal model (Zuffi *et al.*, 2017) and synthetic images. Sanakoyeu *et al.* (2020) explore knowledge distillation (Hinton *et al.*, 2015) to transfer DensePose (Alp Güler *et al.*, 2018) to chimpanzees. These are still the early days of animal pose estimation and we expect many more interesting works in the future.

5) Animal pose estimation discussed so far presents a major challenge that the algorithms must generalise to a multitude of new categories. Unlike the human category, which is represented by an abundance of diverse datasets, animals come in different shapes and forms and it is infeasible to launch extensive data annotation for every single species. This calls for algorithms that can learn without extensive supervision. Similarly to how deep learning revolutionised computer vision by automatically learning features from labeled datasets, the next step on the path to fully automated learning is to enable learning without extensively labeled datasets. Generalising to new categories is not the only obstacle for deploying these algorithms in the real world. Even if one possesses a labeled dataset for a category of interest, the domain gap between training and testing distribution could prevent the deep learning models from generalising properly. This can be partially mitigated by

domain adaptation approaches (Mehta *et al.*, 2017). A viable direction to pursue is using synthetic data (Varol *et al.*, 2017), but it is also unclear how well this would generalise to the in-the-wild setting.

Unsupervised learning of landmarks was pioneered by Thewlis *et al.* (2017) who train a landmark detector that must be equivariant to the basic image transformations. The discovered landmarks are consistent both semantically and geometrically. An alternative approach to discovering object parts is by disentanglement of shape and appearance in an auto-encoder with a structured bottleneck (Lorenz *et al.*, 2019). Unsupervised discovery of landmarks and more generally of semantic parts is receiving more and more attention in the literature (Honari *et al.*, 2018; Suwajanakorn *et al.*, 2018; Xu *et al.*, 2019b; Jakab *et al.*, 2020; Lathuilière *et al.*, 2020). These algorithms are closing the gap with fully supervised training and we believe that they will become essential for successful commercial deployment.

6) In this thesis we developed deep learning based models for human pose estimation and reconstruction of 3D objects and each task was addressed independently. Humans, however, do not exist in isolation from the environment and usually occupy inhabitable spaces as well as interact with 3D objects. The environment imposes constraints on the human pose which can considerably reduce the search space during pose estimation. For example, sitting in a chair, placing hand on a keyboard provides a very rich signal that constraints the pose of a person. The recent work performs test time optimization of 3D human pose while respecting surface inter-penetration constraints (Hassan *et al.*, 2019). Hasson *et al.* (2019) explore joint reconstruction of hand pose and objects during manipulation. The recent line of work explores human-object interaction (HOI) (Monszpart *et al.*, 2019), joint 3D reconstruction of environment and human pose by representing a scene with a parse graph and a Markov Random Field over its terminal nodes (Chen *et al.*, 2019b) and generating plausible human poses respecting scene constraints (Zhang *et al.*, 2020). While these early works achieve promising results, we expect more work in the space of end-to-end holistic scene reconstruction. We also see significant potential in utilising shape reconstruction from vast collections of objects in an unsupervised manner in order to aid estimation of HOI while respecting object affordances. This could benefit ego-centric video analysis where data streams from external and on-body cameras could be progressively combined to perform joint reconstruction.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, *et al.* (2016). TensorFlow: A system for large-scale machine learning, in *OSDI 2016*. 112, 124

K.-A. Aliev, D. Ulyanov, and V. Lempitsky (2020). Neural point-based graphics, *ECCV*. 107

R. Alp Güler, N. Neverova, and I. Kokkinos (2018). Densepose: Dense human pose estimation in the wild, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018*. 127

S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele (2009). Multi-view pictorial structures for 3D human pose estimation, in *BMVC 2009*. 81, 93

A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, *et al.* (2017). A low power, fully event-based gesture recognition system, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. 126

B. Andres (2015). Lifting of Multicuts, *CoRR*, vol. abs/1503.03791. 12

M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele (2018). PoseTrack: A Benchmark for Human Pose Estimation and Tracking, in *CVPR 2018*. 47, 56

M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele (2014). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis, in *CVPR 2014*. 3, 12, 16, 17, 19, 28, 29, 45, 47, 48, 58, 62, 88

M. Andriluka, S. Roth, and B. Schiele (2008). People-Tracking-by-Detection and People-Detection-by-Tracking, in *CVPR 2008*. 14

M. Andriluka, S. Roth, and B. Schiele (2009). Pictorial Structures Revisited: People Detection and Articulated Pose Estimation, in *CVPR 2009*. 9, 13

M. Andriluka, S. Roth, and B. Schiele (2010). Monocular 3D Pose Estimation and Tracking by Detection, in *CVPR 2010*. 14

M. Andriluka, S. Roth, and B. Schiele (2011). Discriminative Appearance Models for Pictorial Structures, *IJCV*. 20

A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt (2011). A Data-driven Approach for Real-time Full Body Pose Reconstruction from a Depth Camera, in *ICCV 2011*. 81

N. Bansal, A. Blum, and S. Chawla (2004). Correlation clustering, *Machine learning*, vol. 56(1-3), pp. 89–113. 22

V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic (2014). 3D pictorial structures for multiple human pose estimation, in *CVPR 2014*. 93

V. Belagiannis and A. Zisserman (2017). Recurrent human pose estimation, in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) 2017*. 10

M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr (2010). A study of parts-based object class detection using complete graphs, *International journal of computer vision*, vol. 87(1-2), p. 93. 9

K. Bernardin and R. Stiefelhagen (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics, *Image and Video Processing*, vol. 2008(1), pp. 1–10. 48, 54

G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani (2019). Learning temporal pose estimation from sparsely-labeled videos, in *Advances in Neural Information Processing Systems 2019*. 15, 70, 126

V. Blanz and T. Vetter (1999). A Morphable Model for the Synthesis of 3D Faces, in *SIGGRAPH 1999*. 90

F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black (2016). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image, in *European Conference on Computer Vision 2016*. 123

C. Bregler and J. Malik (1998). Tracking people with twists and exponential maps, in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231) 1998*. 14, 81

P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman (2008). Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts, in *BMVC 2008*. 9

A. Bulat, J. Kossaifi, G. Tzimiropoulos, and M. Pantic (2020). Toward fast and accurate human pose estimation via soft-gated skip connections, *arXiv preprint arXiv:2002.11098*. 11

A. Bulat and G. Tzimiropoulos (2016). Human pose estimation via Convolutional Part Heatmap Regression, in *ECCV 2016*. 10, 58

M. Burenius, J. Sullivan, and S. Carlsson (2013). 3D Pictorial Structures for Multiple View Articulated Pose Estimation, in *CVPR 2013*. 81

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh (2017). Realtime multi-person 2d pose estimation using part affinity fields, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. 12, 14, 15, 58, 63, 64, 65, 66, 69

J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik (2016). Human Pose Estimation with Iterative Error Feedback, in *CVPR 2016*. 10, 29, 58

J. Carreira and A. Zisserman (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset, in *CVPR 2017*. 15, 67

T. J. Cashman and A. W. Fitzgibbon (2013). What shape are dolphins? Building 3D morphable models from 2D images, *PAMI*, vol. 35. 101

E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion (2005). A survey on participating media rendering techniques, *The Visual Computer*, vol. 21(5), pp. 303–328. 85

J. Chai and J. K. Hodgins (2005). Performance animation from low-dimensional control signals, *ACM Transactions on Graphics*, vol. 24(3), pp. 686–696. 80

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu (2015). ShapeNet: An Information-Rich 3D Model Repository, Technical report arXiv:1512.03012. 112

J. Charles, T. Pfister, D. Magee, and A. Hogg, D. Zisserman (2016). Personalizing Human Video Pose Estimation, in *CVPR 2016*. 14, 16, 18, 58, 67

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, in *ICLR 2015*. 23

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2017a). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *TPAMI*. 9, 65

Q. Chen and V. Koltun (2017). Photographic Image Synthesis with Cascaded Refinement Networks, in *ICCV 2017*. 109

W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler (2019a). Learning to predict 3d objects with an interpolation-based differentiable renderer, in *Advances in Neural Information Processing Systems 2019*. 106

X. Chen and A. Yuille (2014). Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations, in *NIPS 2014*. 10, 13, 20, 29, 82

X. Chen and A. Yuille (2015). Parsing Occluded People by Flexible Compositions, in *CVPR 2015*. 11, 33, 34

Y. Chen, S. Huang, T. Yuan, S. Qi, Y. Zhu, and S.-C. Zhu (2019b). Holistic++ scene understanding: Single-view 3D holistic scene parsing and human pose estimation with human-object interaction and physical commonsense, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 128

Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang (2017b). Adversarial posenet: A structure-aware convolutional network for human pose estimation, in *Proceedings of the IEEE International Conference on Computer Vision 2017*. 10

Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun (2018). Cascaded pyramid network for multi-person pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018*. 13

Z. Chen and H. Zhang (2019). Learning implicit fields for generative shape modeling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 103, 104

B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang (2020). HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. 12

A. Cherian, J. Mairal, K. Alahari, and C. Schmid (2014). Mixing Body-Part Sequences for Human Pose Estimation, in *CVPR 2014*. 14

J. Chibane, T. Alldieck, and G. Pons-Moll (2020). Implicit functions in feature space for 3d shape reconstruction and completion, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 104

W. Choi (2015). Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor, in *ICCV 2015*. 62

C.-J. Chou, J.-T. Chien, and H.-T. Chen (2018). Self adversarial training for human pose estimation, in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2018*. 10

C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese (2016). 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction, in *ECCV 2016*. 102

X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang (2017). Multi-context attention for human pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. 10

J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei (2017). Deformable convolutional networks, in *Proceedings of the IEEE international conference on computer vision 2017*. 15, 69

M. Dantone, J. Gall, C. Leistner, and L. V. Gool. (2013). Human Pose Estimation using Body Parts Dependent Joint Regressors, in *CVPR 2013*. 9, 16

E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica (2006). Correlation clustering in general weighted graphs, *Theoretical Computer Science*. 22

B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi (2020). Cvxnet: Learnable convex decomposition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 105

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database, in *2009 IEEE conference on computer vision and pattern recognition 2009*. 3, 16

A. Doering, U. Iqbal, and J. Gall (2018). Joint flow: Temporal flow fields for multi person tracking. 69, 70

EgoCap (2016). *EgoCap dataset, http://gvv.mpi-inf.mpg.de/projects/EgoCap/*. 80

M. Eichner and V. Ferrari (2010). We Are Family: Joint Pose Estimation of Multiple Persons, in *ECCV 2010*. 11, 16, 17, 32, 33, 49, 53

M. Eichner and V. Ferrari (2012). Appearance Sharing for Collective Human Pose Estimation, in *ACCV 2012*. 9

D. Eigen, C. Puhrsch, and R. Fergus (2014). Depth map prediction from a single image using a multi-scale deep network, in *Advances in neural information processing systems 2014*. 9

A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt (2015). Efficient ConvNet-based Marker-less Motion Capture in General Scenes with a Low Number of Cameras, in *CVPR 2015*. 78, 81, 89, 93

H. Fan, H. Su, and L. J. Guibas (2017). A Point Set Generation Network for 3D Object Reconstruction from a Single Image, in *CVPR 2017*. 100, 103

X. Fan, K. Zheng, Y. Lin, and S. Wang (2015). Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation., in *CVPR 2015*. 29

A. Fathi, A. Farhadi, and J. M. Rehg (2011). Understanding egocentric activities, in *ICCV 2011*. 82

P. F. Felzenszwalb and D. P. Huttenlocher (2005). Pictorial Structures for Object Recognition, *IJCV*. 9, 13

V. Ferrari, M. Marin, and A. Zisserman (2008). Progressive Search Space Reduction for Human Pose Estimation, in *CVPR 2008*. 9, 13

V. Ferrari, M. Marin-Jimenez, and A. Zisserman (2009). Pose search: retrieving people using their pose, in *2009 IEEE Conference on Computer Vision and Pattern Recognition 2009*. 13

M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele (2018). Learning to refine human pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2018*. 11

M. A. Fischler and R. A. Elschlager (1973). The Representation and Matching of Pictorial Structures, *IEEE Trans. Comput'73*. 8

K. Fragkiadaki, H. Hu, and J. Shi (2013). Pose from flow and flow from pose, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013*. 14

J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel (2010). Optimization and Filtering for Human Motion Capture, *IJCV*, vol. 87(1–2), pp. 75–92. 81

G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, *et al.* (2019). Event-based vision: A survey, *arXiv preprint arXiv:1904.08405*. 126

K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser (2020). Local Deep Implicit Functions for 3D Shape, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 105

K. Genova, F. Cole, D. Vlasic, A. Sarna, W. T. Freeman, and T. Funkhouser (2019). Learning shape templates with structured implicit functions, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 104

G. Ghiasi, Y. Yang, D. Ramanan, and C. Fowlkes (2014). Parsing Occluded People, in *CVPR 2014*. 33

R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran (2018). Detect-and-track: Efficient pose estimation in videos, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018*. 15, 70

R. Girdhar, G. Gkioxari, L. Torresani, D. Ramanan, M. Paluri, and D. Tran (2017). Simple, efficient and effective keypoint tracking, in *ICCV PoseTrack Workshop 2017*. 63, 64, 66, 67, 68, 69

R. Girshick (2015). Fast R-CNN, in *ICCV 2015*. 23

R. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, in *CVPR 2014*. 9

G. Gkioxari, A. Toshev, and N. Jaitly (2016). Chained Predictions Using Convolutional Neural Networks. 14, 38, 58

T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry (2018). A papier-mâché approach to learning 3d surface generation, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. 102

S. Günel, H. Rhodin, D. Morales, J. Campagnolo, P. Ramdya, and P. Fua (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila, *Elife*, vol. 8, p. e48571. 127

H. Guo, T. Tang, G. Luo, R. Chen, Y. Lu, and L. Wen (2018). Multi-domain pose network for multi-person pose estimation and tracking, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 70

A. Guzmán-rivera, D. Batra, and P. Kohli (2012). Multiple Choice Learning: Learning to Produce Multiple Structured Outputs, in *NIPS 2012*. 109

S. Ha, Y. Bai, and C. K. Liu (2011). Human motion reconstruction from force sensors, in *SCA 2011*. 80

M. Hassan, V. Choutas, D. Tzionas, and M. J. Black (2019). Resolving 3D human pose ambiguities with 3D scene constraints, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 128

Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid (2019). Learning joint reconstruction of hands and manipulated objects, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 128

K. He, G. Gkioxari, P. Dollár, and R. Girshick (2017). Mask R-CNN, in *ICCV 2017*. 13, 15, 63, 64, 66, 69

K. He, X. Zhang, S. Ren, and J. Sun (2016). Deep Residual Learning for Image Recognition, in *CVPR 2016*. 10, 21, 23, 45, 58, 82, 88

G. Hidalgo, Y. Raaj, H. Idrees, D. Xiang, H. Joo, T. Simon, and Y. Sheikh (2019). Single-Network Whole-Body Pose Estimation, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 12

G. Hinton, O. Vinyals, and J. Dean (2015). Distilling the knowledge in a neural network, *arXiv preprint arXiv:1503.02531*. 127

S. Hochreiter and J. Schmidhuber (1997). Long short-term memory, *Neural computation*, vol. 9(8), pp. 1735–1780. 105

M. B. Holte, C. Tran, M. M. Trivedi, and T. B. Moeslund (2012). Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments, *IEEE Journal of Selected Topics in Signal Processing*, vol. 6(5), pp. 538–552. 81

S. Honari, P. Molchanov, S. Tyree, P. Vincent, C. Pal, and J. Kautz (2018). Improving landmark localization with semi-supervised learning, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018*. 128

P. Hu and D. Ramanan (2016). Bottom-Up and Top-Down Reasoning with Hierarchical Rectified Gaussians, in *CVPR 2016*. 58

J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.* (2016). Speed/accuracy trade-offs for modern convolutional object detectors, *arXiv preprint arXiv:1611.10012*. 48, 63, 65

J. Hwang, J. Lee, S. Park, and N. Kwak (2019). Pose estimator and tracker using temporal flow maps for limbs, in *2019 International Joint Conference on Neural Networks (IJCNN) 2019*. 69, 70

E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele (2017). ArtTrack: Articulated Multi-person Tracking in the Wild, in *CVPR 2017*. 18, 59, 63, 65, 70

E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele (2016a). DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model, in *ECCV 2016*. 15, 38, 39, 42, 44, 45, 51, 53, 58, 63, 65, 66

E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele (2016b). DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model, *arXiv*. 49, 51, 53

S. Ioffe and C. Szegedy (). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *CoRR'15*. 28

C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu (2013). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments, *PAMI*. 18

U. Iqbal and J. Gall (2016). Multi-Person Pose Estimation with Local Joint-to-Person Associations, in *ECCVw 2016*. 14, 51, 58

U. Iqbal, M. Garbade, and J. Gall (2017a). Pose for Action - Action for Pose, in *FG 2017*. 58

U. Iqbal, A. Milan, and J. Gall (2017b). PoseTrack: Joint Multi-Person Pose Estimation and Tracking, in *CVPR 2017*. 15, 16, 18, 55, 56, 59, 63, 65, 70, 123

K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov (2019). Learnable triangulation of human pose, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 126

A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler (2014a). Learning human pose estimation features with convolutional networks. 9, 82

A. Jain, J. Tompson, Y. LeCun, and C. Bregler (2014b). Modeep: A deep learning framework using motion features for human pose estimation, in *Asian conference on computer vision 2014*. 10, 14, 82

T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi (2020). Self-supervised Learning of Interpretable Keypoints from Unlabelled Videos, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 128

H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black (2013). Towards understanding action recognition, in *ICCV 2013*. 16, 18, 58

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell (2014). Caffe: Convolutional Architecture for Fast Feature Embedding, *arXiv preprint arXiv:1408.5093*. 45, 124

H. Jiang and K. Grauman (2016). *Seeing Invisible Poses: Estimating 3D Body Pose from Egocentric Video*. 81

Y. Jiang, D. Ji, Z. Han, and M. Zwicker (2020). Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 107

S. Jin, W. Liu, W. Ouyang, and C. Qian (2019). Multi-person articulated tracking with spatial and temporal embeddings, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 15

S. Jin, X. Ma, Z. Han, Y. Wu, W. Yang, W. Liu, C. Qian, and W. Ouyang (2017). Towards Multi-Person Pose Tracking: Bottom-up and Top-down Methods, in *ICCV PoseTrack Workshop 2017*. 63, 64, 66, 69

S. Johnson and M. Everingham (2009). Combining discriminative appearance and segmentation cues for articulated human pose estimation, in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*. 9

S. Johnson and M. Everingham (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation, in *BMVC'10 2010*. 3, 9, 16, 19, 20, 28

S. Johnson and M. Everingham (2011). Learning Effective Human Pose Estimation from Inaccurate Annotation, in *CVPR 2011*. 16, 28, 82, 88

A. Jones, G. Fyffe, X. Yu, W.-C. Ma, J. Busch, R. Ichikari, M. Bolas, and P. Debevec (2011). Head-Mounted Photometric Stereo for Performance Capture. 81

H. Joo (2019). *Sensing, Measuring, and Modeling Social Signals in Nonverbal Communication*, Ph.D. thesis, Carnegie Mellon University. 1

H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh (2015). Panoptic Studio: A Massively Multiview System for Social Motion Capture, in *ICCV 2015*. 78, 81

H. Kato, Y. Ushiku, and T. Harada (2018). Neural 3D Mesh Renderer, in *CVPR 2018*. 101, 102, 107

L. Ke, M.-C. Chang, H. Qi, and S. Lyu (2018). Multi-scale structure-aware network for human pose estimation, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 11

A. Kendall and Y. Gal (2017). What uncertainties do we need in bayesian deep learning for computer vision?, in *Advances in neural information processing systems 2017*. 12

M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres (2015). Efficient Decomposition of Image and Mesh Graphs by Lifted Multicuts, in *International Conference on Computer Vision 2015*. 44

M. Kiefel and P. Gehler (2014). Human Pose Estimation with Fields of Parts, in *ECCV 2014*. 9

D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier (2012). Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor, in *UIST 2012*. 81

D. P. Kingma and J. Ba (2015). Adam: A Method for Stochastic Optimization, in *ICLR 2015*. 113

K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto (2011). Fast unsupervised ego-action learning for first-person sports videos, in *CVPR 2011*. 82

M. Kocabas, S. Karagoz, and E. Akbas (2018). Multiposenet: Fast multi-person pose estimation using pose residual network, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 12, 14

P. Krähenbühl and V. Koltun (2011). Efficient inference in fully connected crfs with gaussian edge potentials, in *Advances in neural information processing systems 2011*. 10

S. Kreiss, L. Bertoni, and A. Alahi (2019). Pifpaf: Composite fields for human pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 12, 14

A. Krizhevsky, I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks, in *NIPS 2012*. 3, 9, 16, 19

H. W. Kuhn (1955). The Hungarian Method for the assignment problem, *Naval Research Logistics Quarterly*, pp. 83–97. 15

L. Ladicky, P. H. Torr, and A. Zisserman (2013). Human Pose Estimation using a Joint Pixel-wise and Part-wise Formulation, in *CVPR 2013*. 11

C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler (2017). Unite the people: Closing the loop between 3d and 2d human representations, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2017*. 123

S. Lathuilière, S. Tulyakov, E. Ricci, N. Sebe, *et al.* (2020). Motion-supervised Co-Part Segmentation, *arXiv preprint arXiv:2004.03234*. 128

H. Law and J. Deng (2018). Cornernet: Detecting objects as paired keypoints, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 123

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, vol. 86(11), pp. 2278–2324. 9

C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu (). Deeply-supervised nets, in *AISTATS'15* . 24

E. Levinkov, J. Uhrig, S. Tang, M. Omran, E. Insafutdinov, A. Kirillov, C. Rother, T. Brox, B. Schiele, and B. Andres (2017). Joint Graph Decomposition And Node Labeling: Problem, Algorithms, Applications, in *CVPR 2017*. 38, 42, 44, 53, 65

J. Li, K. Xu, S. Chaudhuri, E. Yumer, H. Zhang, and L. Guibas (2017). GRASS: Generative Recursive Autoencoders for Shape Structures, *SIGGRAPH*. 103

C.-H. Lin, C. Kong, and S. Lucey (2018). Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction, in *AAAI 2018*. 101, 103, 113, 114, 115, 136

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár (2017). Focal loss for dense object detection, in *Proceedings of the IEEE international conference on computer vision 2017*. 13, 123

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). Microsoft COCO: Common Objects in Context, in *ECCV 2014*. 12, 16, 17, 58, 70

S. Liu, T. Li, W. Chen, and H. Li (2019a). Soft rasterizer: A differentiable renderer for image-based 3d reasoning, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 106

S. Liu, S. Saito, W. Chen, and H. Li (2019b). Learning to infer implicit surfaces without 3d supervision, in *Advances in Neural Information Processing Systems 2019*. 106

S. Liu, Y. Zhang, S. Peng, B. Shi, M. Pollefeys, and Z. Cui (2020). Dist: Rendering deep implicit signed distance function with differentiable sphere tracing, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 107

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2016a). SSD: Single shot multibox detector, in *ECCV 2016*. 64

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2016b). Ssd: Single shot multibox detector, in *European conference on computer vision 2016*. 123

J. Long, E. Shelhamer, and T. Darrell (2015). Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2015*. 9, 23

M. M. Loper and M. J. Black (2014). OpenDR: An Approximate Differentiable Renderer, in *ECCV 2014*. 101, 106

M. M. Loper, N. Mahmood, and M. J. Black (2014). MoSh: Motion and shape capture from sparse markers, *ACM Transactions on Graphics*, vol. 33(6), pp. 220:1–13. 80

W. E. Lorensen and H. E. Cline (1987). Marching cubes: A high resolution 3D surface construction algorithm, *ACM siggraph computer graphics*, vol. 21(4), pp. 163–169. 103

D. Lorenz, L. Bereska, T. Milbich, and B. Ommer (2019). Unsupervised part-based disentangling of object shape and appearance, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 128

L. Ma and H. Institute (2017). Towards Realtime 2D Pose Tracking: A Simple Online Pose Tracker, in *ICCV PoseTrack Workshop 2017*. 63, 64, 66, 69

M. Ma, H. Fan, and K. M. Kitani (2016). Going Deeper into First-Person Activity Recognition, in *CVPR 2016*. 82

A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning, *Nature neuroscience*, vol. 21(9), pp. 1281–1289. 124, 126

D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision, in *2017 international conference on 3D vision (3DV) 2017*. 128

A. Meka, M. Zollhöfer, C. Richardt, and C. Theobalt (2016). Live Intrinsic Video, *ACM Transactions on Graphics*, vol. 35(4), pp. 109:1–14. 80, 88

A. Menache (2010). *Understanding Motion Capture for Computer Animation*, Morgan Kaufmann, 2nd edn. 78

L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger (2019). Occupancy networks: Learning 3d reconstruction in function space, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 103, 104

M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla (2019). Neural rerendering in the wild, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 107

A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler (2016). MOT16: A Benchmark for Multi-Object Tracking, *arXiv:1603.00831 [cs]*. 62, 124

T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal (Eds.) (2011). *Visual Analysis of Humans: Looking at People*, Springer. 81

A. Monszpart, P. Guerrero, D. Ceylan, E. Yumer, and N. J. Mitra (2019). iMapper: interaction-guided scene mapping from monocular videos, *ACM Transactions on Graphics (TOG)*, vol. 38(4), pp. 1–15. 128

P. Moulon, P. Monasse, and R. Marlet (2013). Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion, in *ICCV 2013*. 96

R. M. Murray, S. S. Sastry, and L. Zexiang (1994). *A Mathematical Introduction to Robotic Manipulation*, CRC Press. 84

S. K. Mustikovela, V. Jampani, S. D. Mello, S. Liu, U. Iqbal, C. Rother, and J. Kautz (2020). Self-Supervised Viewpoint Learning From Image Collections, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 106

N. Neverova and I. Kokkinos (2018). Mass displacement networks. 11

N. Neverova, J. Thewlis, R. A. Guler, I. Kokkinos, and A. Vedaldi (2019). Slim densepose: Thrifty learning from sparse annotations and motion cues, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 70

A. Newell, Z. Huang, and J. Deng (2017). Associative embedding: End-to-end learning for joint detection and grouping, in *Advances in Neural Information Processing Systems 2017*. 12, 15

A. Newell, K. Yang, and J. Deng (2016). Stacked Hourglass Networks for Human Pose Estimation, in *ECCV 2016*. 10, 58, 64, 82

T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang (2019). Hologan: Unsupervised learning of 3d representations from natural images, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 105

T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra (2020). BlockGAN: Learning 3D Object-aware Scene Representations from Unlabelled Images, *arXiv preprint arXiv:2002.08988*. 106

M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger (2020). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 106

G. Ning and H. Huang (2019). Lighttrack: A generic framework for online top-down human pose tracking, *arXiv preprint arXiv:1905.02822*. 69, 70

G. Ning, P. Liu, X. Fan, and C. Zhang (2018). A top-down approach to articulated human pose estimation and tracking, in *European Conference on Computer Vision Workshops 2018*. 70

C. Niu, J. Li, and K. Xu (2018). Im2struct: Recovering 3d shape structure from a single rgb image, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. 103

K. Ohnishi, A. Kanehira, A. Kanezaki, and T. Harada (2016). Recognizing Activities of Daily Living with a Wrist-mounted Camera, in *CVPR 2016*. 82

G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy (2018). Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 12

G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy (2017). Towards Accurate Multi-person Pose Estimation in the Wild, in *CVPR 2017*. 13, 47, 58, 65

D. Park and D. Ramanan (2011). N-best maximal decoders for part models, in *2011 International Conference on Computer Vision 2011*. 14

H. S. Park, E. Jain, and Y. Sheikh (2012). 3D Social Saliency from Head-mounted Cameras, in *NIPS 2012*. 82

J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove (2019). Deepsdf: Learning continuous signed distance functions for shape representation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 103, 104

S. I. Park and J. K. Hodgins (2008). Data-driven Modeling of Skin and Muscle Deformation, *ACM Transactions on Graphics*, vol. 27(3), pp. 96:1–6. 80

D. Paschalidou, L. V. Gool, and A. Geiger (2020). Learning Unsupervised Hierarchical Part Decomposition of 3D Objects from a Single RGB Image, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 103

D. Paschalidou, A. O. Ulusoy, and A. Geiger (2019). Superquadrics revisited: Learning 3d shape parsing beyond cuboids, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 103

C. Payer, T. Neff, H. Bischof, M. Urschler, and D. Stern (2017). Simultaneous Multi-Person Detection and Single-Person Pose Estimation With a Single Heatmap Regression Network, in *ICCV PoseTrack Workshop 2017*. 63, 69

S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger (2020). Convolutional occupancy networks, *arXiv preprint arXiv:2003.04618*. 104

T. D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz (2019). Fast animal pose estimation using deep neural networks, *Nature methods*, vol. 16(1), pp. 117–125. 127

T. Pfister, J. Charles, and A. Zisserman (2015). Flowing ConvNets for Human Pose Estimation in Videos, in *ICCV 2015*. 14, 24, 38, 58, 82

L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele (2013a). Poselet Conditioned Pictorial Structures, in *CVPR 2013*. 9, 13, 20

L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele (2013b). Strong Appearance and Expressive Spatial Models for Human Pose Estimation, in *ICCV 2013*. 9

L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele (2016). DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation, in *CVPR 2016*. 4, 10, 11, 14, 15, 20, 21, 22, 23, 27, 28, 29, 30, 32, 33, 34, 38, 39, 44, 48, 58, 62, 82, 123, 124

G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixé, M. Müller, H.-P. Seidel, and B. Rosenhahn (2011). Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling, in *ICCV 2011*. 80

G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn (2010). Multisensor-fusion for 3D full-body human motion capture, in *CVPR 2010*. 80

G. Pons-Moll, D. J. Fleet, and B. Rosenhahn (2014). Posebits for Monocular Human Pose Estimation, in *CVPR 2014*. 82

Y. Raaj, H. Idrees, G. Hidalgo, and Y. Sheikh (2019). Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 15, 69, 70

U. Rafi, I.Kostrikov, J. Gall, and B. Leibe (2016). An Efficient Convolutional Network for Human Pose Estimation, in *BMVC 2016*. 58

V. Ramakrishna, T. Kanade, and Y. Sheikh (2013). Tracking human pose by tracking symmetric parts, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013*. 14

V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh (2014). Pose Machines: Articulated Pose Estimation via Inference Machines, in *ECCV 2014*. 9

D. Ramanan, D. A. Forsyth, and A. Zisserman (2005). Strike a Pose: Tracking People by Finding Stylized Poses, in *CVPR 2005*. 14

D. Ramanan and C. Sminchisescu (2006). Training deformable models for localization, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) 2006*. 13

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi (2016). You only look once: Unified, real-time object detection, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2016*. 123

S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster R-CNN: Towards real-time object detection with region proposal networks, in *NIPS 2015*. 13, 33, 48, 65, 123

D. Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess (2016). Unsupervised Learning of 3D Structure from Images, in *NIPS 2016*. 101

N. Rhinehart and K. M. Kitani (2016). Learning Action Maps of Large Environments via First-Person Vision, in *CVPR 2016*. 82

H. Rhodin, N. Robertini, D. Casas, C. Richardt, H.-P. Seidel, and C. Theobalt (2016). General Automatic Human Shape and Motion Capture Using Volumetric Contour Cues, in *ECCV 2016*. 84, 86

H. Rhodin, N. Robertini, C. Richardt, H.-P. Seidel, and C. Theobalt (2015). A Versatile Scene Model With Differentiable Visibility Applied to Generative Pose Estimation, in *ICCV 2015*. 81, 85, 89, 93, 101

H. Rhodin, M. Salzmann, and P. Fua (2018). Unsupervised geometry-aware representation for 3d human pose estimation, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 105

S. R. Richter and S. Roth (2018). Matryoshka networks: Predicting 3d geometry via nested shape layers, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2018*. 103

G. Rogez, M. Khademi, J. S. Supancic, III, J. M. M. Montiel, and D. Ramanan (2014). 3D Hand Pose Detection in Egocentric RGB-D Images. 81

O. Ronneberger, P. Fischer, and T. Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation, in *MICCAI 2015*. 11, 104

S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li (2019). Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 104

A. Sanakoyeu, V. Khalidov, M. S. McCarthy, A. Vedaldi, and N. Neverova (2020). Transferring dense pose to proximal animal classes, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 127

B. Sapp and B. Taskar (2013). Multimodal Decomposable Models for Human Pose Estimation, in *CVPR 2013*. 3, 16, 17, 28, 91

B. Sapp, A. Toshev, and B. Taskar (2010). Cascaded Models for Articulated Pose Estimation, in *ECCV 2010*. 9

B. Sapp, D. Weiss, and B. Taskar (2011). Parsing human motion with stretchable models, in *CVPR 2011*. 9, 14, 16

D. Scaramuzza, A. Martinelli, and R. Siegwart (2006). A toolbox for easily calibrating omnidirectional cameras, in *Intelligent Robots and Systems (IROS) 2006*. 86, 87

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. 9

T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins (2011). Motion Capture from Body-mounted Cameras, *ACM Transactions on Graphics*, vol. 30(4), pp. 31:1–10. 78, 80, 94

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake (2011a). Real-Time Human Pose Recognition in Parts from a Single Depth Image, in *CVPR 2011*. 126

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake (2011b). Real-time Human Pose Recognition in Parts from Single Depth Images, in *CVPR 2011*. 81

H. Sidenbladh, M. J. Black, and D. J. Fleet (2000). Stochastic tracking of 3D human figures using 2D image motion, in *European conference on computer vision 2000*. 14

L. Sigal, A. Balan, and M. J. Black (2010). HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion, *International Journal of Computer Vision*, vol. 87. 18, 81, 93

L. Sigal, M. Isard, H. Haussecker, and M. J. Black (2012). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation, *IJCV*, vol. 98(1), pp. 15–48. 81

T. Simon, H. Joo, I. Matthews, and Y. Sheikh (2017). Hand keypoint detection in single images using multiview bootstrapping, in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition 2017*. 126

K. Simonyan and A. Zisserman (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, *CoRR*. 19, 24, 58

V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer (2019a). Deepvoxels: Learning persistent 3d feature embeddings, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 105

V. Sitzmann, M. Zollhöfer, and G. Wetzstein (2019b). Scene representation networks: Continuous 3D-structure-aware neural scene representations, in *Advances in Neural Information Processing Systems 2019*. 105

J. Sivic, M. Everingham, and A. Zisserman (2005). Person spotting: video shot retrieval for face sets, in *International conference on image and video retrieval 2005*. 14

M. Snower, A. Kadav, F. Lai, and H. P. Graf (2020). 15 Keypoints Is All You Need. 15, 69, 70, 124

A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum (2017). Synthesizing 3D Shapes via Modeling Multi-View Depth Maps and Silhouettes with Deep Generative Networks, in *CVPR 2017*. 103

S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt (2015). Fast and Robust Hand Tracking Using Detection-Guided Optimization, in *CVPR 2015*. 79, 81

C. Stoll, N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt (2011). Fast articulated motion tracking using a sums of Gaussians body model, in *ICCV 2011*. 81

Y.-C. Su and K. Grauman (2016). Detecting Engagement in Egocentric Video, in *ECCV 2016*. 82

Y. Sugano and A. Bulling (2015). Self-Calibrating Head-Mounted Eye Trackers Using Egocentric Visual Saliency, in *UIST 2015*. 81

K. Sun, B. Xiao, D. Liu, and J. Wang (2019). Deep high-resolution representation learning for human pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 11, 13, 69

X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman (2018). Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling, in *CVPR 2018*. 113

S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning, in *Advances in neural information processing systems 2018*. 128

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning., in *AAAI 2017*. 48, 65

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going Deeper with Convolutions, in *CVPR 2015*. 24

C. Szegedy, A. Toshev, and D. Erhan (2013). Deep neural networks for object detection, in *Advances in neural information processing systems 2013*. 9

S. Tang, B. Andres, M. Andriluka, and B. Schiele (2015). Subgraph Decomposition for Multi-Target Tracking, in *CVPR 2015*. 15, 38, 39, 41, 42, 44

S. Tang, B. Andres, M. Andriluka, and B. Schiele (2016). Multi-Person Tracking by Multicuts and Deep Matching, in *BMTT 2016*. 46

W. Tang, P. Yu, and Y. Wu (2018a). Deeply learned compositional models for human pose estimation, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 11

Z. Tang, X. Peng, S. Geng, L. Wu, S. Zhang, and D. Metaxas (2018b). Quantized densely connected u-nets for efficient landmark localization, in *Proceedings of the European Conference on Computer Vision (ECCV) 2018*. 11

M. Tatarchenko, A. Dosovitskiy, and T. Brox (2017). Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs, in *ICCV 2017*. 102

J. Tautges, A. Zinke, B. Krüger, J. Baumann, A. Weber, T. Helten, M. Müller, H.-P. Seidel, and B. Eberhardt (2011). Motion Reconstruction Using Sparse Accelerometer Data, *ACM Transactions on Graphics*, vol. 30(3), pp. 18:1–12. 80

B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua (2016). Direct Prediction of 3D Body Poses from Motion Compensated Sequences, in *CVPR 2016*. 82

C. Theobalt, E. de Aguiar, C. Stoll, H.-P. Seidel, and S. Thrun (2010). Performance capture from multi-view video, in R. Ronfard and G. Taubin (eds.), *Image and Geometry Processing for 3-D Cinematography 2010*, pp. 127–149, Springer. 81

J. Thewlis, H. Bilen, and A. Vedaldi (2017). Unsupervised learning of object landmarks by factorized spatial embeddings, in *Proceedings of the IEEE international conference on computer vision 2017*. 128

R. Tokola, W. Choi, and S. Savarese (2013). Breaking the chain: liberation from the temporal Markov assumption for tracking human poses, in *ICCV 2013*. 14

J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler (2015). Efficient Object Localization Using Convolutional Networks, in *CVPR 2015*. 9, 23, 29, 58

J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler (2014). Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation, in *NIPS 2014*. 9, 10, 20, 23, 24, 28, 29, 58, 82, 91

A. Toshev and C. Szegedy (2014). DeepPose: Human Pose Estimation via Deep Neural Networks, in *CVPR 2014*. 9, 10, 28, 58, 82

D. Tran and D. A. Forsyth (2010). Improved Human Parsing with a Full Relational Model, in *ECCV 2010*. 9

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon (1999). Bundle adjustment—a modern synthesis, in *International workshop on vision algorithms 1999*. 105

S. Tulsiani, A. A. Efros, and J. Malik (2018). Multi-view Consistency as Supervisory Signal for Learning Shape and Pose Prediction, in *CVPR 2018*. 102, 113, 116, 117, 118, 119, 121, 132, 136

S. Tulsiani, A. Kar, J. Carreira, and J. Malik (2017a). Learning Category-Specific Deformable 3D Models for Object Reconstruction, *PAMI*, vol. 39. 101

S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik (2017b). Learning Shape Abstractions by Assembling Volumetric Primitives, in *CVPR 2017*. 102

S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik (2017c). Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency, in *CVPR 2017*. 100, 101, 102, 111, 113, 114, 115, 122, 136

R. Urtasun, D. J. Fleet, and P. Fua (2006). Temporal motion models for monocular and multiview 3D human body tracking, *Computer Vision and Image Understanding (CVIU)*, vol. 104(2), pp. 157–177. 81

G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid (2017). Learning from Synthetic Humans, in *CVPR 2017*. 67, 128

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need, in *Advances in neural information processing systems 2017*. 15, 71

T. Verelst and T. Tuytelaars (2020). Dynamic Convolutions: Exploiting Spatial Sparsity for Faster Inference, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 126

S. Vicente, J. Carreira, L. Agapito, and J. Batista (2014). Reconstructing PASCAL VOC, in *CVPR 2014*. 101

D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović (2007). Practical motion capture in everyday surroundings, *ACM Transactions on Graphics*, vol. 26(3), p. 35. 80

C. Vondrick, D. Patterson, and D. Ramanan (2012). Efficiently Scaling up Crowdsourced Video Annotation, *IJCV*. 60

B. Wandt and B. Rosenhahn (2019). Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 11

F. Wang and Y. Li (2013). Beyond Physical Connections: Tree Models in Human Pose Estimation., in *CVPR 2013*. 9

J. Wang, Y. Cheng, and R. S. Feris (2016). Walk and Learn: Facial Attribute Representation Learning from Egocentric Video and Contextual Data, in *CVPR 2016*. 81

M. Wang, J. Tighe, and D. Modolo (2020). Combining detection and tracking for human pose estimation in videos, in *CVPR 2020*. 15, 69, 70, 71, 124

R. Y. Wang and J. Popović (2009). Real-time hand-tracking with a color glove, *ACM Transactions on Graphics*, vol. 28(3), p. 63. 80

S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh (2016). Convolutional Pose Machines, in *CVPR 2016*. 10, 12, 20, 23, 24, 28, 29, 58, 82

X. Wei, P. Zhang, and J. Chai (2012). Accurate Realtime Full-body Motion Capture Using a Single Depth Camera, *ACM Transactions on Graphics*, vol. 31(6), pp. 188:1–12. 81

P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid (2013). DeepFlow: Large displacement optical flow with deep matching, in *ICCV 2013*. 46

D. J. Weiss and B. Taskar (2013). Learning Adaptive Value of Information for Structured Prediction, in *NIPS 2013*. 14

O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson (2020). Synsin: End-to-end view synthesis from a single image, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 107

J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman (2018). 3D Interpreter Networks for Viewer-Centered Wireframe Modeling, *IJCV*. 103

J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling, in *NIPS 2016*. 102

J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, *et al.* (2017). AI challenger: A large-scale dataset for going deeper in image understanding, *arXiv preprint arXiv:1711.06475*. 16, 17

S. Wu, C. Rupprecht, and A. Vedaldi (2020). Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 107

F. Xia, P. Wang, X. Chen, and A. L. Yuille (2017). Joint multi-person pose estimation and semantic part segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017*. 13

B. Xiao, H. Wu, and Y. Wei (2018). Simple baselines for human pose estimation and tracking, in *Proceedings of the European conference on computer vision (ECCV) 2018*. 15, 69, 70

Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu (2018). Pose flow: Efficient online pose tracking. 69

L. Xu, W. Xu, V. Golyanik, M. Habermann, L. Fang, and C. Theobalt (2020). EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 126

Q. Xu, W. Wang, D. Ceylan, R. Mech, and U. Neumann (2019a). DISN: Deep implicit surface network for high-quality single-view 3d reconstruction, in *Advances in Neural Information Processing Systems 2019*. 104

Z. Xu, Z. Liu, C. Sun, K. Murphy, W. T. Freeman, J. B. Tenenbaum, and J. Wu (2019b). Unsupervised discovery of parts, structure, and dynamics, in *Proceedings of the International Conference on Learning Representations 2019*. 128

X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee (2016). Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision, in *NIPS 2016*. 100, 101, 105, 114, 115, 136

B. Yang and R. Nevatia (2012). An Online Learned CRF Model for Multi-Target Tracking, in *CVPR 2012*. 62

W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang (2017). Learning feature pyramids for human pose estimation, in *proceedings of the IEEE international conference on computer vision 2017*. 11

Y. Yang, C. Feng, Y. Shen, and D. Tian (2018). FoldingNet: Interpretable Unsupervised Learning on 3D Point Clouds, in *CVPR 2018*. 102, 122

Y. Yang and D. Ramanan (2012). Articulated Human Detection with Flexible Mixtures of Parts, *PAMI*. 9, 11, 20, 82

H. Yasin, U. Iqbal, B. Krüger, A. Weber, and J. Gall (2016). A Dual-Source Approach for 3D Pose Estimation from a Single Image, in *CVPR 2016*. 82

K. Yin and D. K. Pai (2003). Footsee: an interactive animation system, in *SCA 2003*. 80

H. Yonemoto, K. Murasaki, T. Osawa, K. Sudo, J. Shimamura, and Y. Taniguchi (2015). Egocentric articulated pose tracking for action recognition, in *International Conference on Machine Vision Applications (MVA) 2015*. 81

D. Yu, K. Su, J. Sun, and C. Wang (2018). Multi-person Pose Estimation for Pose Tracking with Enhanced Cascaded Pyramid Network, in *European Conference on Computer Vision Workshops 2018*. 70

D. Zhang and M. Shah (2015). Human pose estimation in videos, in *Proceedings of the IEEE International Conference on Computer Vision 2015*. 14

P. Zhang, K. Siu, J. Zhang, C. K. Liu, and J. Chai (2014). Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture, *ACM Transactions on Graphics*, vol. 33(6), pp. 221:1–14. 81

S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, and S.-M. Hu (2019). Pose2seg: detection free human instance segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2019*. 16, 17

W. Zhang, M. Zhu, and K. G. Derpanis (2013). From actemes to action: A strongly-supervised representation for detailed action understanding, in *CVPR 2013*. 16, 18, 58

Y. Zhang, M. Hassan, H. Neumann, M. J. Black, and S. Tang (2020). Generating 3D People in Scenes without People, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020*. 128

S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr (2015). Conditional random fields as recurrent neural networks, in *Proceedings of the IEEE international conference on computer vision 2015*. 10

X. Zhou, J. Zhuo, and P. Krahenbuhl (2019). Bottom-up object detection by grouping extreme and center points, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019*. 123

X. Zhu, Y. Jiang, and Z. Luo (2017). Multi-Person Pose Estimation for PoseTrack with Enhanced Part Affinity Fields, in *ICCV PoseTrack Workshop 2017*. 63, 64, 66, 69

C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem (2017). 3d-prnn: Generating shape primitives with recurrent neural networks, in *Proceedings of the IEEE International Conference on Computer Vision 2017*. 102

S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black (2019). Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture From Images, in *Proceedings of the IEEE International Conference on Computer Vision 2019*. 127

S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black (2017). 3D menagerie: Modeling the 3D shape and pose of animals, in *Proceedings of the IEEE conference on computer vision and pattern recognition 2017*. 127

S. Zuffi, J. Romero, C. Schmid, and M. J. Black (2013). Estimating human pose with flowing puppets, in *proceedings of the IEEE International Conference on Computer Vision 2013*. 14

PUBLICATIONS

5. E. Insafutdinov, A. Dosovitskiy. Unsupervised Learning of Shape and Pose with Differentiable Point Clouds. In *Advances in Neural Information Processing Systems* (2018)

4. H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele and C. Theobalt. EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras. In *ACM Transactions on Graphics* (2016)

3. M. Andriluka, U. Iqbal, E. Insafutdinov, L. Pishchulin, A.Milan, J. Gall and B. Schiele. PoseTrack: A Benchmark for Human Pose Estimation and Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition* (2018)

2. E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres and B. Schiele. Arttrack: Articulated multi-person tracking in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (oral, 2.65% acceptance rate)* (2017)

1. E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka and B. Schiele. DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. In *European Conference on Computer Vision* (2016)