UNIVERSITÄT BONN

Articulated Human Pose Estimation in Unconstrained Images and Videos

Dissertation

zur

Erlangung des Doktorgrades (*Dr. rer. nat.*)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich–Wilhelms–Universität, Bonn

vorgelegt von

Umar IQBAL

aus

Lahore, Pakistan

Bonn 2018

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachter: Advisor: Prof. Dr. Juergen Gall
2. Gutachter: Prof. Dr. Vincent Lepetit
Tag der Promotion: 30.11.2018
Erscheinungsjahr: 2018

Abstract

by Umar Iqbal

for the degree of

Doctor rerum naturalium

The understanding of the articulated human body pose is of great interest in many scenarios. While humans have an unmatched ability to effortlessly extract and interpret such information in any unconstrained environment, developing computational methods with similar capabilities is a very challenging task. The developed methods have to handle scenes with complex backgrounds, an unknown number of potentially occluded and truncated people, large-scale variations, diverse lighting conditions, and the vast amounts of appearance variation due to complex body articulations and clothing. The noise introduced by the lossy sensing modalities complicates the problem even further. While there has been a lot of work for human pose estimation in constrained environments, very few works have addressed these challenges in the literature. Further, the estimation of the articulated pose of small functional body parts such as "hands" has often been ignored in the existing works. To this end, this thesis addresses the aforementioned challenges and presents efficient and robust computational methods for the 2D and 3D articulated human body and hand pose estimation in unconstrained real-world scenarios.

First, we address the problem of 2D multi-person body pose estimation. We present an efficient approach that estimates the poses of people in groups or crowd. We demonstrate that the problem can be formulated as a set of local joint-to-person association problems which can be solved efficiently for each person in the image, while also handling occlusions and truncations.

Second, we introduce the challenging case of simultaneous multi-person pose estimation and tracking in videos. The approaches for multi-person pose estimation in images cannot be applied directly to this problem since it also requires to solve person associations over time. To this end, we propose a novel method that jointly models both problems in a single formulation using a spatio-temporal graph. The optimization of the graph using integer linear programming directly provides plausible body pose trajectories for each person. The proposed method does not make any assumptions and performs pose estimation and tracking in fully unconstrained videos. We also present a large scale dataset and a thorough evaluation protocol to evaluate the developed methods quantitatively. Further, we provide an extensive analysis of the performance of state-of-the-art methods and highlight their strengths and weaknesses.

Given the estimated, possibly noisy, 2D pose trajectory of a person, the third direction of this thesis focuses on the refinement of pose trajectory by exploiting the information about human activities. We present an action-conditioned pictorial structure model that predicts and incorporates activity information for body pose refinement.

The fourth direction of this thesis concerns 3D human pose estimation from single images. Given the estimated 2D pose of a person, we present an approach to lift the 2D pose to

3D by using an efficient and robust method for 3D pose retrieval and reconstruction. Unlike existing works, the proposed approach does not require any training images with annotated 3D poses. Since we can estimate 2D poses from any unconstrained image, the proposed method can also reconstruct 3D poses in any unconstrained scenario.

The final part of the thesis concerns the estimation of 3D hand pose from an RGB input. We present a novel 2.5D pose representation which can be estimated reliably from an RGB image and allows to reconstruct the absolute 3D pose of the hand using a novel 3D reconstruction approach. The proposed method can handle severe occlusions, complex hand articulations, and unconstrained images taken from the wild.

Keywords: articulated pose estimation, multi-person pose tracking, human body pose, hand pose, 2D to 3D, 3D reconstruction

Dedicated to my family.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Juergen Gall, for giving me an excellent opportunity to work under his supervision. I am thankful for his unwavering support, exciting ideas, and high quality feedback on my work, and nuturing me as a reseacher. I am also grateful to him for always providing me what was required to accomplish the goals, may it be computational resources, hiwis, or financial funding for annotations. I have never come back from his office without having a solution to my problems.

I would also like to thank Prof. Vincent Lepetit for immediately agreeing to serve as an external reviewer, and Prof. Andreas Weber and Prof. Gabriel Schaaf for being part of the thesis committee.

I am grateful to my colleagues in Bonn for making my Ph.D. journey fun and full of learning. I am thankful to Abhilash Srikantha for helping me during my move to Bonn, and to endless discussions, we had during our time together in Bonn. I want to thank Dimitris Tzionas for always being there for consultation and tips. Abhilash and Dimitris have always been great mentors throughout my Ph.D. I am grateful to Martin Garbade for courteously sharing the office with me for three years, for interesting discussions, and most importantly for doing an excellent job on managing the computational resources of the group. I am grateful to Alexander Richard for great memories during our trips abroad for conferences, and for his awesome queuing tool to manage the resources. I want to thank Hilde Kuehne for interesting discussions and tips on thesis writing, and also for proof-reading parts of this thesis. I am grateful to Andreas Doering for always believing in me as his supervisor, and for great collaborations on several projects. Finally, I would like to thank the second generation of our group; Johann Sawatzky, Sovan Biswas Ahsan Iqbal, Yaser Souri, Mohsen Fayyaz, Rania Briq, and Yazan Abu Farha. I have always been impressed by their skills and motivation to do foundational research. I am also thankful to Muhammad Omer Saeed for a fruitful collaboration during his master's thesis.

I would like to express my sincere gratitude to all my collaborators. I am grateful to Anton Milan for his successful collaboration on PoseTrack. I am also grateful to the extended PoseTrack family from MPII Saarbrucken; Misha Andriluka, Leonid Pishculin, Eldar Insafutdinov, and Prof. Bernt Schiele. I had a fantastic time at NVIDIA Research as an intern, thanks to the mentorship and guidance of Jan Kautz and Pavlo Molchanov. A massive shout-out to Learning and Perception team at NVIDIA Research for embellishing my internship experience.

I would also like to thank my former advisor Saquib Sarfraz. After all, it all started when he supervised my bachelor thesis, and later hired me as a research associate in his lab.

Having a right place to live is crucial while living away from home. I am thankful to the very dear friend (and landlord) Faraz Dahar for providing a home away from home, and making me part of his family. I couldn't have asked for more. I am also grateful to my cousin sister Aisha Omer and her husband Omer Sheikh for their unconditional help at difficult times.

I would like to thank my parents and siblings for their never-ending support throughout my life. It is the result of the confidence and sense of security that they have given me that I was able to pursue my interest. Without them, this thesis would never have happened.

Doing a Ph.D. can at times be hard on your personal life. I want to thank my beloved wife Mehru for her unconditional love, wholehearted support, selfless sacrifice, and taking good care of our Son, Muhammad, at times I was away working late on this thesis. None of this would been possible

without her support. Muhammad was born during the first year of my Ph.D., and ever since he made sure that I did not bring any work home, or even if I did, I was not able to do that. Thank you, Muhammad, for making the life after work meaningful and joyous.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

An alphabetically sorted list of abbreviations used in the thesis:

2d/3d	Two-dimensional/Three-dimensional
AI	Artificial Intelligence
AP	Average Precision
AUC	Area Under the Curve
BP	Belief Propagation
CPM	Convolutional Pose Machines
CRF	Conditional random field
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
DT	Dense Trajectories
DPM	Deformable Part Model
EM	Expectation Maximization
FM	Fragements
FV	Fisher Vector
GMM	Gaussian mixture model
HOG	Histogram of Oriented Gradients
IDT	Improved Dense Trajectories
ID	Identity switches
IoU	Intersection over Union
ILP	Integer Linear Program
KS	Keypoint Similarity
LBP	Loopy belief propagation
LJPA	Local Joint-to-Person Association
LP	Linear Program
LSTM	Long Short-Term Memory
mAP	mean Average Precision
MAP	Maximum-a-Posteriori
MoCap	Motion Capture
MOT	Multi-Object Tracking
MOTA	Multi-Object Tracking Accuracy
MOTP	Multi-Object Tracking Precision
ML	Mostly Lost
mm	millimeters
MT	Mostly Tracked
OKS	Object Keypoint Similarity
PCK	Percentage of Correct Keypoints

PCKh	Percentage of Correct Keypoints after head normalization
PS	Pictorial Structure
px	Pixel
RBF	Radial basis function
RNN	Recurrent Neural Network
RGB-D	RGB-Depth
SGD	Stochastic gradient descent
SVM	Support vector machine

Frequently Used Symbols

$\setminus$	Set subtraction operation
$\mathbf{I}$	Image
$\mathcal{F}$	Video sequence
f	Video frame
F	Total number of frames in a video sequence
$\mathbf{x}$	2D location $\mathbf{x} = (x, y)$
$\mathcal{X}$	Set of all pixel locations
$\mathbf{X}$	3D location $\mathbf{X} = (X, Y, Z)$
j	Index of the body joint/keypoint
$\mathcal{J}$	Set of all body joints or keypoints
J	Number of joints/keypoints
d	A detection candidate
D	A set of detection candidates
$\mathbf{p}$	2D pose
$\mathbf{P}$	3D pose
$\mathcal{G}$	Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
$\mathcal{V}$	Vertices of graph $\mathcal{G}$
$\mathcal{E}$	Edges of graph $\mathcal{G}$
$\mathcal{M}$	Projection matrix
$\mathcal{K}$	Intrinsic camera parameters
E	Energy function
$\mathcal{L}$	Loss function
p	Probability
$\phi(\cdot)$	Unary potential
$\psi(\cdot, \cdot)$	Pairwise potential
θ	Parameters
T	A random tree
$\mathcal{T}$	A random forest
a	Action label
$\mathcal{A}$	Set of all action labels
A	Total number of action labels
$\mathbf{g}$	feature vector/descriptor

g	type of feature descriptor
ω	2D pose space
Ω	3D pose space
H	Heatmap or confidence scoremap

List of Publications

- H. Yasin, U. Iqbal, B. Krueger, A. Weber, and J. Gall
  A Dual-Source Approach for 3D Pose Estimation from a Single Image
  In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
  http://pages.iai.uni-bonn.de/iqbal_umar/ds3dpose/

- U. Iqbal and J. Gall.
  Multi-Person Pose Estimation with Local Joint-to-Person Associations
  In European Conference on Computer Vision (ECCV) Workshops, Crowd Understanding (CUW), 2016.
  http://pages.iai.uni-bonn.de/iqbal_umar/multiperson-pose/

- U. Iqbal, M. Garbade and J. Gall.
  Pose for Action - Action for Pose
  In IEEE Conference on Automatic Face and Gesture Recognition (FG), 2017.
  http://pages.iai.uni-bonn.de/iqbal_umar/action4pose/

- U. Iqbal, A. Milan and J. Gall.
  PoseTrack: Joint Multi-Person Pose Estimation and Tracking
  In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
  http://pages.iai.uni-bonn.de/iqbal_umar/PoseTrack/
  Video: https://youtu.be/SgiFPWNuAGw

- M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall and B. Schiele
  PoseTrack: A Benchmark for Human Pose Estimation and Tracking
  In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
  http://posetrack.net
  Video: https://youtu.be/uYFRxGyMDe4

- U. Iqbal, A. Doering, H. Yasin, B. Krueger, A. Weber, and J. Gall
  A Dual-Source Approach for 3D Human Pose Estimation from Single Images
  In Computer Vision and Image Understanding (CVIU), 2018.
  http://pages.iai.uni-bonn.de/iqbal_umar/ds3dpose/

- U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz
  Hand Pose Estimation via Latent 2.5D Heatmap Regression
  In European Conference on Computer Vision (ECCV), 2018.
  Video: https://youtu.be/4Q3ByHZ8tNc

CHAPTER 1

Introduction

Fie, fie upon her!
There's language in her eye, her cheek, her lip,
Nay, her foot speaks; her wanton spirits look out
At every joint and motive of her body.

William Shakespeare,
Troilus and Cressida

Contents

1.1. MOTIVATION

Recent advances in Artificial Intelligence (AI) have resulted in the integration of an ever-increasing number of AI technologies in our daily lives. One of the quests in this direction is to develop autonomous systems that can integrate into our daily lives and interact with us as naturally as another human could do. Humans interact with each other and with their environment in complex ways. Even if we do not speak, our bodies transmit a lot of information such as our behavior, intention, or activity via hand and body gestures, facial expressions and body movements. While the humans can effortlessly detect and interpret such information from subtle and complex body signals, we need to endow these autonomous systems with similar capabilities to integrate them naturally in real-world human environments. Supporting such detailed reasoning would eventually need highly sophisticated

computer vision systems – able to estimate human body pose, hand articulations, facial landmarks, and perceive their motion, all in the three-dimensional world space.

Besides, the developed methods can also be utilized in numerous practical applications. For example, the information about the body pose can be used to recognize the activity of the persons, and to detect abnormal behaviors in surveillance systems. A self-driving car can anticipate the intentions of the pedestrians or cyclists by looking at their body and hand pose to make timely decisions. A driver-assistance system can alert the sleeping or distracted drivers by looking at their body pose and facial landmarks. The articulation of the fingers can be used for sign-language or hand-gesture recognition, or in touch-less natural interfaces for extended reality (AR/VR/MR) scenarios. In the entertainment industry, body pose information can be used for gaming, or to replace expensive motion capture systems to apply special effects without the need for specialized suits. Further, service robots can use body pose information in assisting people who need help with their day-to-day tasks or interactions *e.g.*, elderly people or children with developmental disabilities.

However, to be applicable in complex real-world scenarios, the developed methods must operate under varied background and lighting conditions, scenes with an unknown number of persons, sever body part occlusions and truncations, and large appearance variations exhibited by the human body. Further, they should be robust against noise due to sensing modalities such as motion blur, depth and scale ambiguities, low image resolution, and compression artifacts. The small functional parts of the human body, such as hands, present additional challenges due to the small size and heavy occlusions.

While the problem of articulated human pose estimation has been studied for a long time in the literature, most of the earlier works make strong assumptions and ignore the aforementioned challenges posed by the realistic scenarios. For example, a large number of earlier works for 2D body pose estimation assumes that only a single, pre-localized person is visible in the image (Felzenszwalb and Huttenlocher, 2005; Peng et al., 2018). For 3D body pose estimation, the used training data is often recorded in indoor settings. Hence, the trained methods are only applicable in similar constrained scenarios (Sminchisescu et al., 2005; Zhou et al., 2016b). In contrast to body pose estimation, hand pose estimation has received much less attention in the literature and only a few works exist that address this challenging problem from only RGB images (Simon et al., 2017; Zimmermann and Brox, 2017; Mueller et al., 2018).

The goal of this thesis, therefore, is to address the challenges mentioned above and fill the gaps in the literature. To this end, in this thesis, we focus on developing computational methods for articulated pose estimation of humans from completely unconstrained images and videos. We start by presenting a method for 2D body pose estimation of multiple, potentially truncated and occluded, people. As a natural next step, we then move to the problem of multi-person pose estimation and tracking in videos. The methods developed for pose estimation in images cannot be applied directly to this problem since in this case persons should also be associated over time. To this end, we present an approach that jointly models pose estimation and tracking in a single formulation. Given the 2D pose trajectories of a person, we then propose a method to refine the pose trajectories by exploiting the information about human activities. Subsequently, given the estimated 2D poses, we then address the problem of lifting the 2D poses to 3D. For this, we present an efficient and robust method for 3D pose retrieval and reconstruction. Finally, we also address the challenging problem of hand pose estimation from RGB images and present an approach which can work on images taken from the wild.

In the following, we describe the problem formulation used in this thesis, followed by a brief

description of the contributions made by the thesis in Section 1.3.

1.2. PROBLEM FORMULATION

We describe the pose by the locations of J number of keypoints[1] predefined using an anatomical structure, and develop methods for both 2D and 3D pose estimation. We define the 2D pose as $\mathbf{p} = \{\mathbf{x}_j\}_{j\in\mathcal{J}}$ and 3D pose as $\mathbf{P} = \{\mathbf{X}_j\}_{j\in\mathcal{J}}$, where $\mathbf{x}_j = (x_j, y_j) \in \mathbb{R}^2$ represents the 2D pixel coordinates of the body keypoint j in image $\mathbf{I}$ and $\mathbf{X}_j = (X_j, Y_j, Z_j) \in \mathbb{R}^3$ denotes the location of the keypoint in the 3D camera coordinate frame measured in millimeters.

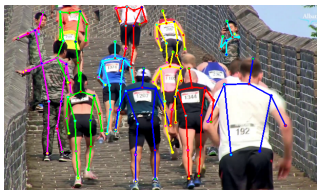
The level of detail and complexity depends on the application. The detail of the pose may depend on the number of keypoints to be localized and their dimensionality (*i.e.*, 2D or 3D). In the case of video data, we may also need to track the person(s) and assign them unique identities. The complexity may depend on the environment *i.e.*, indoor setting with a single person or outdoor scene with complex backgrounds and multiple interacting people. The problems can be simplified further by assuming sophisticated multi-camera setup or depth camera. For applications such as photo-realistic social VR or medical diagnosis, we may require very dense landmarks, but of a single person with a simple background. In contrast, for sports video analytics or surveillance systems, the locations of a sparse set of body keypoints might be sufficient, but the number of persons and background cannot be restricted.

In this thesis, we do not make any assumption on the environment or the number of persons, and assume RGB input captured using a single camera, for both 2D and 3D pose estimation. We use a pose representation consisting of a sparse set of anatomical landmarks *i.e.*, $J \in [13, 15]$ for body pose, and $J = 21$ for hand pose. Such a sparse representation has been shown to be sufficient for the human visual system to recognize the activity (Johansson, 1973), gender (Kozlowski and Cutting, 1977), identity (Perrett et al., 1985), and sign-language (Poizner et al., 1981).

1.3. CONTRIBUTIONS

In this thesis we make the following contributions towards articulated pose estimation.

1.3.1. Multi-Person 2D Pose Estimation in Images



We start by proposing an efficient method that estimates the body poses of multiple persons in an image in which a person can be occluded by another person or might be truncated. To this end, we consider multi-person pose estimation as a joint-to-person association problem. We construct a fully connected graph from a set of detected joint candidates in an image and resolve the joint-to-person association and outlier detection using integer linear programming. Since solving joint-to-person association jointly for all persons in an image is an NP-hard problem and even approximations are expensive, we demonstrate that the problem can be addressed instead by solving a set of local joint-to-person association problems. Our method

[1] In this thesis, we will interchangeably use the words: keypoint, joint, and part. In any case, they correspond to some region on the human body.

runs at a significantly better runtime while achieving an accuracy similar to the competing methods. This part is based on the work published in (Iqbal and Gall, 2016).

1.3.2. Joint Multi-Person Pose Estimation and Tracking

As a natural next step, we address the problem of multi-person pose estimation and tracking in videos. Existing methods for multi-person pose estimation in images cannot be applied directly to this problem, since it also requires to solve the problem of person association over time in addition to the pose estimation for each person. In this thesis, we formally introduce this challenging problem, and present a new dataset with sixty fully-annotated videos along with a proper evaluation protocol. Further, we also propose a novel method that jointly models multi-person pose estimation and tracking in a single formulation. To this end, we build on our work for multi-person pose estimation in images described above, and represent body joint detections in a video by a spatio-temporal graph. We solve an integer linear program to partition the graph into sub-graphs that correspond to plausible body pose trajectories for each person. The proposed approach implicitly handles occlusion and truncation of the persons, and can work on videos taken from the wild. This part is based on the work published in (Iqbal et al., 2017b).

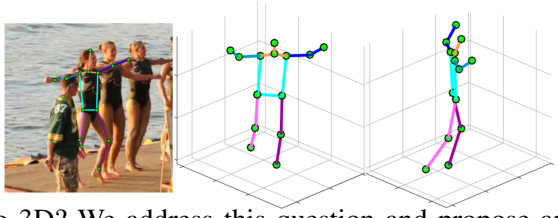
1.3.3. A Benchmark for Human Pose Estimation and Tracking

The break-through developments in computer vision are mainly due to the deep learning methods and availability of large-scale datasets. The thirty (out of sixty) videos used in the work above are, however, insufficient to train sophisticated models or to properly evaluate the developed methods in representative real-world settings. To alleviate this hurdle, we introduce the PoseTrack benchmark which is a new large-scale benchmark for video-based human pose estimation and articulated tracking. To this end, we extend our dataset discussed above to more than 500 videos. We also conduct an extensive experimental study on the recent approaches for multi-person pose estimation and tracking and provide an analysis of the strengths and weaknesses of the state-of-the-art. This part is based on the work published in (Andriluka et al., 2018).

1.3.4. Action Priors for Human Pose Estimation

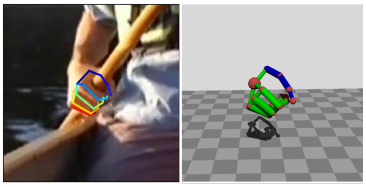
The body pose trajectories, obtained for example from the method discussed above, provide strong cues about the ongoing activity of the person. However, intuitively, the information about the activity can also provide a strong cue about the pose. We exploit this observation and propose to refine the estimated poses by utilizing the information about human actions. To this end, we present a pictorial structure model that incorporates higher-order part dependencies by modeling action specific appearance models and pose priors, where the action priors are obtained using the initial estimates of the poses. This is achieved by starting with a uniform action prior and updating the action prior during pose estimation. We demonstrate the effectiveness of the proposed method on two challenging datasets for pose estimation and action recognition. This part is based on the work published in (Iqbal et al., 2017a).

1.3.5. 3D Body Pose Estimation



Many applications such as assistive robots, gaming, or human-computer interaction also need to know body pose information in 3D. All of the methods introduced so far in this thesis, however, focus only on 2D pose estimation. A natural question then is, can we lift the estimated 2D poses to 3D? We address this question and propose an approach which reconstructs the 3D body pose from the estimated 2D pose, where the 2D pose can be obtained using any of the methods discussed above. In contrast to recent approaches that learn deep neural networks to regress 3D pose directly from images, our proposed method does not require training images with 3D pose annotations. This has the advantage that the proposed method is not restricted to the views only present in the training data, which often consist of the views only from the controlled indoor settings. Hence, our method can also perform pose estimation in unconstrained scenes. We achieve this by decomposing the problem into two subproblems of 2D pose estimation and 3D pose retrieval based on the estimated 2D pose. Such a decomposition utilizes two separate sources of training data including 2D pose annotations and motion capture data, both of which are available abundantly. This part is based on the work published in (Yasin et al., 2016; Iqbal et al., 2018a).

1.3.6. 2D and 3D Hand Pose Estimation



In this last work, we shift our attention to the largely unaddressed problem of hand pose estimation from RGB data. Similar to the approach for 3D human body pose estimation discussed above, we decompose the problem of 3D pose estimation into two subproblems. However, this time we decompose it into the regression of a 2.5D pose representation and the reconstruction of the 3D pose from 2.5D. The 2.5D pose representation encodes the information about the location of hand keypoints in the image and their normalized depths relative to the root keypoint. We design our novel 2.5D pose representation such that it is invariant to scale and translation; it allows to exploit image information effectively; it allows us to perform multi-task learning so that multiple sources of training data can be used, and enables us to devise an approach to exactly recover the absolute 3D pose up to a scaling factor. We also propose a novel CNN architecture to efficiently regress the 2.5D pose from images without the loss of spatial resolution. This is done by learning 2.5D heatmaps in a latent way by using a differentiable loss function. Our proposed approach and the novel network architecture achieve the state-of-the-art accuracy for 2D and 3D hand pose estimation on several challenging datasets. This part is based on the work published in (Iqbal et al., 2018b).

1.4. THESIS STRUCTURE

The thesis is organized as follows. The next chapter provides an overview of the literature for articulated human pose estimation. The third chapter revises the preliminary concepts used in this thesis. While Chapter 4-9 correspond to the contributions described in Section 1.3.

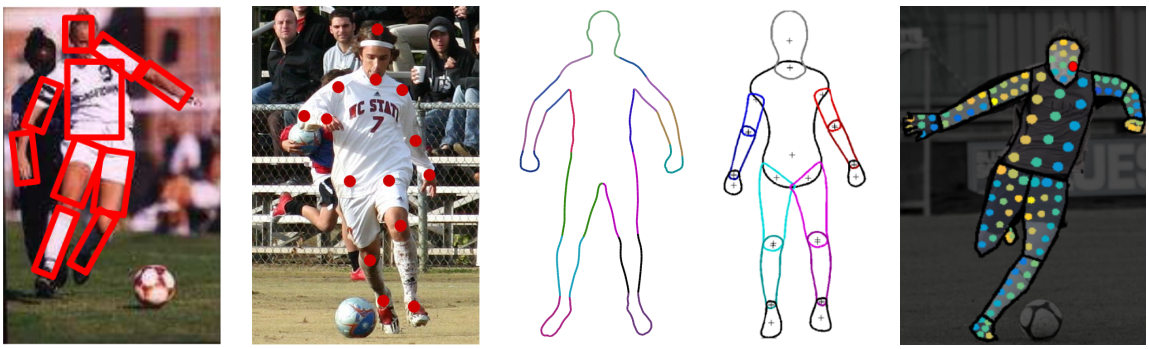
- **Chapter 2** gives a detailed overview of the state-of-the-art for human pose estimation. It starts by reviewing the works that focus only on 2D pose estimation of single, pre-localized persons, and explores different pose representations used in the literature. It then moves to the approaches that also tackle images with multiple persons, and subsequently, reviews the approaches that have been proposed for multi-person pose estimation and tracking. Finally, it provides a detailed overview of the approaches for 3D pose estimation.
- **Chapter 3** provides a brief review of the basic concepts required to understand the developed methods in this thesis.
- **Chapter 4** presents an efficient approach for multi-person pose estimation from RGB images containing multiple interacting people.
- **Chapter 5** introduces the challenging problem of joint multi-person pose estimation and tracking in videos. It presents a novel method that jointly models multi-person pose estimation and tracking in a single formulation. The proposed method estimates body pose trajectories for any unknown number of persons, and can handle occlusion, truncation, and temporal association within a single formulation.
- **Chapter 6** presents PoseTrack, which is a new large-scale benchmark for video-based joint multi-person pose estimation and tracking. It also conducts an extensive experimental study to analyse the state-of-the-art approaches.
- **Chapter 7** presents an action conditioned pictorial structure model that utilizes the information about the activity of the persons to improve body pose estimation. The action information is obtained from the initial estimates of the body poses.
- **Chapter 8** proposes an approach that lifts 2D human poses to 3D. The proposed method does not require training data with labeled 3D pose annotations, but instead relies on two abundantly available sources of training data.
- **Chapter 9** shifts the attention toward hand pose estimation. It presents a new method for 3D hand pose estimation from an RGB image via a novel 2.5D pose representation and a reconstruction approach to recover 3D pose from 2.5D.
- Finally, **Chapter 10** concludes the thesis and provides interesting future directions.

CHAPTER 2

State-of-the-Art

Contents

The visual understanding of human body pose is a long-standing problem and has been studied for decades with different level of granularities. The earliest of the works date back at least to the work by the Swedish Psychologist Gunnar Johansson (Johansson, 1973). In his seminal work, Johansson demonstrated that the motion of a sparse set of body joints is sufficient for humans to understand motion patterns and infer human actions. Johansson's work sparked a huge interest in the computer vision community for automatic understanding and localization of human body posture from image or video data. Starting from the $80s$ (O'rourke and Badler, 1980; Hogg, 1983), the topic encompasses a vast variety of literature (Gavrila, 1999; Forsyth et al., 2006; Moeslund et al., 2011; Sarafianos et al., 2016; Gong et al., 2016; MPII-Leaderboard, 2014; PoseTrack-Leaderboard, 2018), but it is still considered very challenging, in particular, in unconstrained scenarios. The proposed methods can be classified into a variety of categories, for example, based on input modalities (*e.g.*, RGB, Depth or RGB-D), camera setup (*i.e.*, single-view or multi-view), the dimensionality of the output space (2D or 3D), and body shape representation. While the literature for human pose estimation is diverse and massive, in this thesis we only focus on the 2D and 3D pose estimation from a single-view RGB input. For completeness, in this chapter, we will review the approaches with various pose representations, however, in the rest of this thesis, we will focus only on joint-based pose representations. Moreover, in this thesis, we address the problems of the human body pose estimation and hand pose estimation. Generally, both problems are treated separately in the literature. However, they share several properties with each other, and many approaches proposed for the human body can be easily adapted for hand pose estimation or vice-versa. Hence, in the following, we discuss the related works for articulated pose estimation in general.



(a) (b) (c) (d) (e)

Figure 2.1: Examples of body pose representation used in the literature. (a) The limbs-based representation used in (Felzenszwalb and Huttenlocher, 2005). (b) The joint-based representation used in (Yang and Ramanan, 2011; Sapp et al., 2011). (c) The contour people representation proposed by Freifeld et al. (2010). (d) The deformable structures representation of Zuffi et al. (2012). (e) The dense pose representation proposed by Güler et al. (2018). The images are taken from (Felzenszwalb and Huttenlocher, 2005; Yang and Ramanan, 2011; Freifeld et al., 2010; Zuffi et al., 2012; Güler et al., 2018)

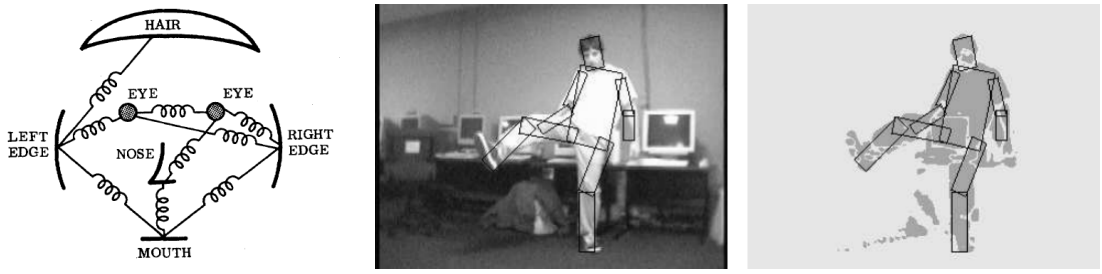
2.1. 2D POSE ESTIMATION

The estimation of 2D human pose from RGB images or videos is a challenging problem in computer vision, mainly due to the complex articulations of the human body, person or object interactions, body part occlusion and truncation, and large amounts of appearance variation due to variable clothing or lightening. There exists a variety of body pose representations used in the literature (see Figure 2.1) such as limb-based representation (Felzenszwalb and Huttenlocher, 2005), joint-based representation (Dantone et al., 2014), contour person (Freifeld et al., 2010), deformable structures (Zuffi et al., 2012), and more recently, the dense pose representation (Güler et al., 2018). For a very long time, the problem of human pose estimation was studied only for single, pre-localized, and fully-visible persons. However, more recent works also address the problem in unconstrained images or videos with multiple interacting people. In this chapter, we first review the approaches focusing on single-person human pose estimation (Section 2.1.1) followed by the review of more recent approaches for multi-person human pose estimation in images (Section 2.1.2) and videos (Section 2.1.3), respectively.

2.1.1. Single-Person-Pose-Estimation

2.1.1.1. Graphical models based methods

The breakthrough developments in pose estimation models started with the seminal work of Felzenszwalb and Huttenlocher (2005) in which they used the Pictorial Structure (PS) model of Fischler and Elschlager (1973) and provided an efficient inference method. The PS model formulates the problem using a tree-structured graphical model with nodes representing the body parts and edges defining the spatial relationships between adjacent body parts. In (Felzenszwalb and Huttenlocher, 2005), the body parts are represented using rectangular templates, while the spatial interactions of the parts are modeled using geometric features based on their relative deformations. The goal of



(a) (b)

Figure 2.2: (a) The pictorial structure model of (Fischler and Elschlager, 1973). (b) Pictorial structure model of (Felzenszwalb and Huttenlocher, 2005). The images are taken from (Fischler and Elschlager, 1973; Felzenszwalb and Huttenlocher, 2005).

the PS model is to find the location, scale, and orientation of each body part such that they are coherent with the image evidence while also satisfying the kinematic constraints of the human body. This is performed by using Max-Product Belief Propagation (BP) to find the maximum-a-posteriori (MAP) solution. The original PS model (Fischler and Elschlager, 1973) performed Max Product BP in quadratic time using dynamic programming. The approach in (Felzenszwalb and Huttenlocher, 2005) instead provided a linear time solution by modeling the spatial pairwise terms with a parametric quadratic function and utilizing an efficient distance transform (Felzenszwalb and Huttenlocher, 2012).

Stronger appearance models. While the PS model model in (Felzenszwalb and Huttenlocher, 2005) provides efficient inference, it relies on simple background subtraction to segment the body parts from the background. Performing the background subtraction in realistic images with complex environments is, however, very challenging. A large number of subsequent works, therefore, focused on to improve the body part segmentation by richer appearance modeling.

Ramanan (2007) proposed an iterative parsing approach that iteratively extracts image-specific features to refine part segmentation. It performs soft labeling of pixels into a body part type at each iteration and uses those label maps to generate foreground/background masks specific to each body part type. The foreground/background masks are then used as additional features in the next parsing iteration to obtain better part labelings. The approach in (Ferrari et al., 2008) extends (Ramanan, 2007) by adopting a progressive search space reduction strategy. They incorporate a generic person detector and GrabCut (Rother et al., 2004) based foreground segmentation to prune invalid part candidates. The approach (Andriluka et al., 2009) incorporated strong discriminatively trained part detectors that provide dense pixel-wise likelihoods for each body part type. They utilized richer shape context descriptors (Belongie et al., 2001) with strong Adaboost classifiers (Haykin et al., 2005) which resulted in significantly improved performance. Pishchulin et al. (2013a) extended the approach of (Andriluka et al., 2012) by augmenting it with rotation specific and rotation invariant part detectors.

All of the aforementioned works use a limb-based pose representation (Figure 2.1a), where each limb corresponds to a part in PS model. The used rectangular part templates for appearance modeling are parametrized for position, scale, and orientation of the limbs. In general, the limbs exhibit large appearance variation due to different clothing, length of the limbs, viewpoints, foreshortening, occlusions, and orientations. This makes it harder for part detectors to detect body parts in uncon-

strained scenarios. To this end, Yang and Ramanan (2011) and Sapp et al. (2011) proposed to use a joint-based pose representation (Figure 2.1b). The positions of body joints are more robust to appearance variations and eliminate the need of explicitly modeling the scale and orientation information used in limb-based models. In fact, a limb with any length or orientation can be implicitly detected by separately detecting both joints forming the limb. The part templates, in this case, are squared boxes around the joint location. Yang and Ramanan (2011) also detected the mid-point of each limb to achieve more flexibility in the PS model and both approaches (Yang and Ramanan, 2011; Sapp et al., 2011) showed great benefits in using a joint-based pose representation. Yang and Ramanan (2011) also propose to use mixture models for joint detectors to achieve more robustness to joint orientations. Eichner and Ferrari (2012) extended the same approach and proposed to incorporate color information to obtain the mixture models. Instead of using explicit mixture models, Dantone et al. (2013) use random forest-based part detectors that are inherently multi-class and can implicitly handle multi-modal data.

Higher order information such as co-occurrence and relations of body parts has also been exploited in the literature to obtain better part detectors. The approach (Eichner and Ferrari, 2009) observed that the relative locations of some body parts (*e.g.*, face and torso) remain stable and often the color distributions of some body parts are statistically related. They exploit this observation and learn better appearance models by learning such relationships in a latent manner. The approaches in (Dantone et al., 2013; Ramakrishna et al., 2014) propose multi-layer part detectors where the first layer uses features extracted from the input image and provides likelihood maps for each body part. In addition to the image features, the second layer also utilizes the output of the first layer as features. This allows the second layer to implicitly capture the information about co-occurrence and interdependencies of the body parts, hence, results in stronger part detectors.

Stronger spatial models. The efficient inference for a PS model is mainly due to the tree-structured graph model and geometric modeling of pairwise spatial terms. However, this formulation has several limitations. First, the tree-structured graph can only model the dependencies between adjacent body parts, therefore, cannot exploit the inherent dependencies between non-adjacent body parts. This limitation often results in the well-known phenomenon of double-counting for symmetric body parts. Secondly, simple geometric features are a too crude representation for pairwise modeling considering the substantial variation of human pose, and cannot exploit richer appearance dependencies between parts. Moreover, the PS model in (Felzenszwalb and Huttenlocher, 2005) assumes that the relative offsets between parts are normally distributed. This is necessary for efficient inference using the distance transform. However, in reality, these offsets have a multi-modal distribution, hence, the models with Gaussian distribution cannot capture complex human articulations. Many works also developed stronger spatial models to improve the performance of PS model further while coping with these challenges.

Johnson and Everingham (2010b) cluster the body poses based on articulation and learn a separate PS model for each cluster, where each PS model specializes for a particular range of body articulations. During inference, several poses are estimated using all PS models and the one with the highest MAP solution is chosen. This, however, increases the computational cost linearly with the number of PS models. Yang and Ramanan (2011) propose to use the Guassian Mixture Model (GMM) by instead clustering the relative offsets and using a Gaussian model for each cluster. During inference, the model that provides the highest likelihood for each pair of location candidates is chosen. This approach allows using exponentially many PS models without having to perform the in-

ference repetitively. Dantone et al. (2014) demonstrate that combining both strategies lead to further improvements.

Other methods also proposed image dependent pairwise terms to leverage the appearance information between body parts. Using image dependent pairwise terms, however, does not allow efficient inference using the distance transform, as done in (Felzenszwalb and Huttenlocher, 2005). To this end, the approach (Sapp et al., 2010) follows a search space reduction approach by using a cascade of coarse-to-fine structured models. It filters a large number of unlikely locations by using simple features that allow efficient inference, and extracts richer appearance features such as segmentation, contours, and shape from finer image scales, but only from a sparse set of remaining location candidates. Pishchulin et al. (2013b) propose a mixture model approach similar to (Yang and Ramanan, 2011) but select the appropriate model for each input image based on poselet detectors. This allows utilizing higher-order part and appearance dependencies while retaining the efficient inference of PS model.

2.1.1.2. Deep Learning based methods

All of the works discussed so far rely on hand-crafted features for appearance modeling. With the breakthrough work of Krizhevsky et al. (2012) on learned feature representations using CNNs (LeCun et al., 1998), many works also employed CNNs for human pose estimation.

Jain et al. (2014a) learned CNN based body part detectors. Each part detector takes an image patch as input and provides the likelihood for the presence of corresponding body part inside that patch. They relied on a simple graphical model to enforce the kinematic constraints of the human body for further improvements. Chen and Yuille (2014) extended PS model by introducing CNN based part-detectors and image-dependent pairwise terms. Instead of processing the images in a sliding-window manner, Tompson et al. (2014b) adopted a fully-convolutional neural network that directly produces heatmaps for each body part, where each heatmap encodes the likelihood of a part being present at a particular pixel. The network consists of two branches each of which processes the input image with different resolution to capture multi-resolution information. Overlapping receptive fields are introduced to exploit richer contextual information. The produced heatmaps are finally processed via an MRF based spatial model to enforce global pose consistency and constrain joint inter-connectivity. The authors also provide an approach to optimize the parameters of the spatial model and CNN in a unified training setup which led to further improvements.

While the aforementioned works still use graphical models to enforce human body constraints, later developments showed that graphical models are of little importance in the presence of robust part detectors since the long-range relationships of the body parts can be directly incorporated in the part detectors trained using CNNs in an end-to-end setup. The so-called DeepPose approach by Toshev and Szegedy (2014) cast the problem as the holistic regression of joint locations. Given an input image, the network processes it via a set of sequential convolutional layers followed by two fully-connected layers and produces the joint-locations as output. The network applies pooling after every convolutional layer to capture the increasing amount of contextual information and to reduce the computational complexity. While this strategy allows capturing the articulation of the human body, it reduces the network's capability to precisely localize the body joints. To cope with this, the authors incorporate additional stages of the CNN to refine the output sequentially. Carreira et al. (2016) proposed an iterative feedback approach that starts with a mean pose and in each iteration,

the network predicts corrections to the current estimate of the pose. The same approach was later adopted by Sun et al. (2017b) where they instead use a ResNet (He et al., 2016) based backbone network for improved performance.

While holistic regression-based approaches showed great improvements, they still suffered from the loss of spatial information and struggled to localize body joints accurately. Therefore, state-of-the-art approaches for 2D human pose estimation rely on the fully-convolutional networks and produce heatmaps as output rather than directly regressing the coordinates of joints. Wei et al. (2016) build on a similar idea as (Dantone et al., 2013; Ramakrishna et al., 2014) and proposed a multi-stage CNN architecture where each stage of the network takes as input the heatmaps of all parts from its preceding stage and the receptive field of the subsequent stages is increased to the extent that the context of the complete person is available. This provides additional information about the interdependence, co-occurrence, and context of parts to each stage, and thereby allows the network to implicitly learn image dependent spatial relationships between parts. The resulting multi-stage network is prone to the vanishing gradients due to a very large depth, therefore, intermediate supervision is added to strengthen the gradients during training. Newell et al. (2016) proposed the stacked-hourglass network architecture where each stage takes the form of an hourglass. Each hourglass network first encodes the input image by lower resolution feature maps using repetitive convolutions and poolings. The the low-resolution feature maps are then decoded using repetitive up-sampling and convolutions to recover the spatial resolution and to generate the joint heatmaps. The encoding part helps the network to learn higher-level semantic representations of the human body by looking at wider contextual information, while the decoding part recovers the spatial resolution and combines the higher-level semantic information with low-level spatial information via skip-connection to exploit richer information. Bulat and Tzimiropoulos (2016) also adopted a multi-staged network architecture and demonstrated similar improvements. Instead of a multi-staged architecture, Insafutdinov et al. (2016) propose to use a very deep network that inherently results in large receptive fields and therefore allows to use contextual information around the parts. They use a ResNet architecture (He et al., 2016) with dilated convolutions (Chen et al., 2017a) to have a higher output resolution. Belagiannis and Zisserman (2017) replace the multi-staged network with a recurrent CNN that tries to iteratively refine the pose estimates by also utilizing the predictions from previous iterations.

More recent methods build on the idea of multi-staged network architectures and propose additional strategies to improve the performance. Sun et al. (2017a) propose a multi-stage framework with local and global pose normalization modules. The global normalization module rotates the joint heatmaps to have an upright position of the body. The normalized heatmaps are then refined via a refinement network which is trained for people with normalized body orientation. The local refinement stage further rotates the refined heatmaps for each body part to have a vertical downward position of the limbs. This is followed by another refinement network which is trained for limbs with normalized orientation. The final pose predictions are obtained by applying the inverse of the local and global rotations on refined heatmaps, respectively.

Lifshitz et al. (2016) observed that most pixels belonging to the person do not lie on the body joints and therefore do not contribute much to the pose estimation process. To exploit information from the entire person, they devised a voting scheme where each pixel votes for the relative location of body joints. The locations with maximum votes are then chosen as the final prediction. Chu et al. (2017b) incorporate richer contextual information by introducing a CRF based attention mechanism

which is designed to learn the spatial correlation between body parts while focusing on the regions of interest. Further, a modified version of the stack-hourglass network (Newell et al., 2016) is proposed that captures multi-resolution, multi-semantic, and hierarchical information. Similarly, Yang et al. (2017) also capture multi-resolution details via a novel pyramid residual module to achieve robustness to scale variations.

Human pose estimation and semantic part segmentation are two related problems in computer vision, and both can provide complementary information to each other. In this direction, Nie et al. (2018b) proposed to exploit the information from semantic human part-segmentation to improve articulated human pose estimation. The proposed network architecture consists of two branches, each responsible for human pose estimation and body part segmentation. The branch for human pose estimation also exploits the features tailored for part segmentation to utilize high-level semantic information.

Peng et al. (2018) improve the performance of Newell et al. (2016) by learning to generate more effective data augmentation during training in an adversarial setup. The training consists of two networks; the augmentation network to generate augmentation operators and a target network that estimates the human body pose given an input image. The augmentation network explores the weaknesses of the pose estimation network and generates hard augmentation operators, while the pose estimation network learns from hard augmentations to achieve better performance. The resulting pose estimation network is eventually trained on well-augmented training data, hence, results in better generalization across different scenarios. Chen et al. (2017b) incorporate a discriminator network with an adversarial loss that distinguishes whether the produced pose is anatomically plausible or not. This provides additional regularization to the backbone network (stacked-hourglass (Newell et al., 2016) in this case) and prevents it from generating implausible poses.

Memory and resource efficient networks for human pose estimation have also been proposed in the literature. While Rafi et al. (2016) adopted a fully-convolutional version of GoogLeNet (Ioffe and Szegedy, 2015) which is more memory efficient than other network architectures, Adrian Bulat (2017) proposed a binarized convolutional network which is $6\times$ more memory efficient than its real-valued version. They demonstrated that a sufficiently good pose estimation accuracy could be achieved with binarized networks that can run on limited resources. However, it is important to note that there is still a performance gap between the binary and real-valued networks (78.1% vs 85.5%).

2.1.1.3. Video based methods

Single person pose estimation in videos has also been studied extensively in the literature. While any single-person pose estimation method for images can be employed here, the primary goal, in this case, is to improve per-frame pose estimations by exploiting the temporal information present in videos, such as by enforcing temporal smoothing constraints and/or by using optical flow information. The earlier methods treated it as a tracking problem where the pose in the first frame was manually initialized and subsequently tracked in the remaining video (Ju et al., 1996). Recent methods, however, follow a tracking-by-detection approach which eliminates the need of manual initialization.

Sapp et al. (2011) formulate the problem as a spatiotemporal graphical model where the body parts are spatially connected via a tree-structured graph but the parts with the same type are also connected temporally. This, however, introduces loops in the graph which makes the model intractable.

To this end, they present an approximate solution by decomposing the loopy graph into an ensemble of tractable tree-structured sub-models, each of which is responsible for tracking one body part while also utilizing the information from neighboring parts. Park and Ramanan (2011) and Batra et al. (2012) generate several pose hypotheses for each video frame using (Yang and Ramanan, 2011) and select a smooth configuration of poses over time from all hypotheses. Instead of complete articulated pose, Ramakrishna et al. (2013) track individual body parts and regularize the trajectories of the body parts through the location of neighboring parts. They jointly track symmetric parts to avoid double counting by enforcing mutual exclusion constraints. Cherian et al. (2014) extend the approach of Park and Ramanan (2011) and select smooth configuration of limbs instead of complete body poses. Zhang and Mubarak (2015) also formulate a spatiotemporal graph that is similar to (Sapp et al., 2011), but also spatially connects the symmetric body parts. The connections between symmetric parts are useful to avoid double-counting, however, they introduce additional cycles in the graph and result in an NP-Hard optimization problem. To this end, Zhang and Mubarak (2015) decompose the graph into two tree-based optimization problems, each of which can be optimized efficiently with exact inference.

Some approaches also perform an instance-specific adaptation of appearance models. Ramanan et al. (2005) assume that the pose of the person can be estimated reliably in at least one frame of the video. They utilize this reliable pose to build a person-specific appearance model and use this to estimate the poses in the rest of the video. A similar iterative approach is adopted in (Shen et al., 2014). In each iteration, they estimate the poses in the entire video and refine them by introducing spatiotemporal smoothness constraints. The high confidence poses from the testing video are automatically selected and used to fine-tune the pose estimation model for the next iterations. More recently, Charles et al. (2016) adopted a similar approach while using a CNN based pose estimation approach.

The motion of the body parts provides powerful visual cues to understand articulated body pose (Johansson, 1973). Therefore, many works also try to use motion features to improve body pose estimation. Jain et al. (2014b) extended the CNN based approach of Tompson et al. (2014b) by also providing additional motion features as input. By combining visual and motion features such as optical flow, they were able to outperform the existing methods. Pfister et al. (2015) warp the heatmaps from neighboring frames using optical flow and combine them with the heatmaps from the current video frame using a convolutional layer to obtain more robust part detections. Song et al. (2017) also adopt a similar approach. Given a thin-slice of the input video, they first estimate the heatmaps of each video frame. These heatmaps are subsequently propagated to the neighboring frames using optical flow information. The original and the warped heatmaps are then passed to a spatiotemporal inference algorithm that produces temporally coherent poses as output.

Recurrent neural networks have also been used to exploit temporal information for pose estimation in videos. Gkioxari et al. (2016) proposed to use a convolutional RNN (Recurrent Neural Network). At each time step, the network also receives the output and the hidden-state from the previous time step as input. Hence it can exploit the temporal dependencies of preceding frames to make the predictions. Luo et al. (2018) extend the approach of Gkioxari et al. (2016) by instead using memory augmented RNN or commonly referred as LSTM (Long Short-Term Memory) networks.

2.1.1.4. *Body shape based methods*

The methods discussed so far in this section either use a limb-based or a joint-based pose representation. Both representations, however, do not provide any information about the shape of the person, *i.e.*, they cannot be used to distinguish between fat or a thin person. To this end, Freifeld et al. (2010) proposed the Contour Person (CP) (Figure 2.1c) model that also captures the shape of the human body. CP is a generative model and is parameterized for pose, shape and camera view. During inference, the parameters of the model are optimized to best describe the contours of the person in the image. CP is a holistic body model and does not allow independent deformations of individual body parts, therefore, it cannot efficiently capture complex articulations of the human body. Zuffi et al. (2012) combined the strengths of traditional PS model and CP in a single model, that they refer to as the Deformable Structure Model (DSM) (Figure 2.1d). DSM allows independent deformations of the body parts and also captures their shape, while also keeping tree-based efficient inference of PS model. More recently, Güler et al. (2018) propose the DensePose representation which requires to localize dense key-points on the human body (Figure 2.1e). They cast the problem as the classification of body regions to UV-coordinates of a 3D body model, the so-called Skinned Multi-Person Linear (SMPL) model (Loper et al., 2015).

2.1.2. Multi-Person Pose Estimation

Most approaches discussed so far in this chapter assume that only a single person is visible in the image, and cannot handle realistic cases where several people appear in the scene, and interact with each other. In contrast to single person pose estimation, multi-person pose estimation introduces significantly more challenges, since the number of persons in an image is not known a priori. Moreover, it is natural that persons occlude each other during interactions, and may also become partially truncated to various degrees. Multi-person pose estimation has, therefore, gained much attention recently. The proposed approaches can be categorized either to the top-down or bottom-up category. We will review both categories in the following.

2.1.2.1. *Top-down methods*

The top-down methods employ person detectors to first localize the persons and then use a single-person pose estimation model to estimate the pose of each person individually. To this end, Pishchulin et al. (2012) use an off-the-shelf PS model based pose estimator (Andriluka et al., 2012). Such an approach, however, is applicable only if people appear well separated and do not occlude each other since the traditional PS model can fail in the presence of multiple interacting persons. Eichner and Ferrari (2010) extend the PS model to explicitly model people interactions by incorporating occlusion priors, where the occlusion probabilities are obtained using a separate occlusion predictor. Additionally, an exclusion penalty term is introduced that prevents body parts of different people from occupying the same image location.

Ladicky et al. (2013) and Yang and Ramanan (2013) adopt a slightly different approach. Instead of relying on a person detector, they extend the approach of Park and Ramanan (2011) and generate N-Best pose candidates that are different in scale or their root joints are sufficiently different. Ladicky et al. (2013) further iteratively prune the implausible pose candidates by measuring their compatibility with the body-part segmentation masks, texture, and color information. The approach

was shown to work well under partial occlusions of the human body. One of the drawbacks of PS model is that it always outputs a fixed number of body joints and does not account for occlusion and truncation, which often is the case in multi-person scenarios. Same is also true for CNN based single person pose estimation methods (Wei et al., 2016; Newell et al., 2016). To handle such cases, Chen and Yuille (2015) explore a large range of tree-structure graphs, each with a different number of body parts and body part types. During inference, an occlusion term is incorporated in the objective function of PS model which penalizes graph-structures containing the occluded body parts. The inference is performed using all of the possible graph structures, and the one with the highest MAP solution is chosen. Since the number of possible graph structures is enormous, an efficient inference method that exploits shared computations between different graph structures is utilized.

More recent top-down approaches also follow a similar strategy but instead use CNN based person detectors (Redmon and Farhadi, 2017; Ren et al., 2015; Liu et al., 2016) and pose estimation models. Papandreou et al. (2017) and Xiao et al. (2018) employ a ResNet (He et al., 2016; Chen et al., 2017a) based pose estimation model with Faster-RCNN (Ren et al., 2015) for person detection. Chen et al. (2018) rely on a stronger person detector of Lin et al. (2017b) and propose a modified version of the stacked-hourglass network (Newell et al., 2016) for pose estimation. The part detectors are trained with online hard-keypoint mining to explicitly improve the performance for body joints that are difficult to localize due to occlusion (*e.g.*, wrists) or clothing variation (*e.g.*, hips and shoulders).

Most of the methods for single-person pose estimation assume that the person of interest lies at the center of the input image. A slight mislocalization of the person by the detectors can, therefore, result in inaccurate pose estimation. To this end, Fang et al. (2017) propose an architecture with symmetric spatial transformer networks. Given a bounding box of the person, the first network applies an affine transformation such that the person lies at the center with an upright position. After performing pose estimation, the second network maps the pose back to the input image.

He et al. (2017) extended Faster-RCNN (Ren et al., 2015) to simultaneously perform person detection and pose estimation. Faster-RCNN consists of a region proposal network that generates class-agnostic object proposals. The object proposals are subsequently passed to a bounding box recognition branch that classifies each bounding box to an object class. He et al. (2017) added a branch for predicting body pose in parallel with the existing branch for bounding box recognition. This allowed them to train the network for the complete pipeline in an end-to-end manner. The approach reported impressive results on the benchmark datasets.

2.1.2.2. Bottom-up methods

The top-down methods have the advantage that they do not need to handle the large amount of scale variation since the person bounding boxes can be normalized to a unit-scale before performing body pose estimation. However, the major drawback of these approaches is that they are prone to early commitment, *i.e.*, if the person detector fails to localize a person, which often is the case when people appear in close proximity, the pose of the missed person can never be estimated. Moreover, the runtime of these methods is linear to the number of persons present in the image. In contrast, bottom-up methods first localize all body joints in the image and then employ an approach for associating them with unique individuals. Bottom-up approaches provide more robustness to early commitment and a runtime that is agnostic to the number of persons in the image. Moreover, in contrast to the top-down approach where each person is tackled separately, bottom-up approaches can exploit the

global contextual cues from other people and other body parts.

In this direction, Pishchulin et al. (2016) are the first to propose a bottom-up approach. They proposed a joint objective function to perform pose estimation of all persons in a single formulation. The proposed method does not require a separate person detector or any prior information about the number of persons. Unlike earlier works, it can tackle any type of occlusion or truncation. The approach starts by generating a set of class independent part proposals and constructs a densely connected graph from the proposals. The graph is optimized using integer linear programming to label each proposal with a particular body part type and assigns them to unique individuals. The optimization problem proposed in (Pishchulin et al., 2016) is theoretically well-founded, but is an NP-Hard problem to solve and prohibitively expensive for realistic scenarios. Insafutdinov et al. (2016) build on (Pishchulin et al., 2016) and used ResNet (He et al., 2016) based stronger part detectors and an image dependent pairwise term along with an incremental optimization approach that significantly reduces the optimization time of (Pishchulin et al., 2016). However, the runtime remained in order of minutes per image. In a follow-up work (Insafutdinov et al., 2017), they further simplified and sparsified the graph and leveraged a faster inference method (Levinkov et al., 2017) to vastly improve the runtime.

The pairwise terms are crucial for the accurate association of body parts in bottom-up approaches. Insafutdinov et al. (2016, 2017) regress offset vectors from every part location to all other parts, and the bi-directional agreement between the offsets of two body parts is considered as the measure of association. Regressing accurate offset vectors is, however, a difficult task. Furthermore, an additional requirement of the logistic regressors to convert the obtained evidence to probabilities further adds to the noise. Therefore, the methods in (Insafutdinov et al., 2016, 2017) had to rely on complex densely connected graphs to combine global evidence to make the final decisions. In contrast, Cao et al. (2017) presented a novel representation of association scores via Part Affinity Fields (PAF). PAF represent the location and orientation of limbs using a set of 2D vectors encoded in an image. For this, they extended the model of Wei et al. (2016) by introducing an additional branch which predicts PAF for all pairs of joints. The joint pairs are fixed based on an almost tree-like graph structure. The authors demonstrated that PAFs provide sufficient evidence of part association that a simple greedy bipartite graph matching algorithm can be used to achieve high-quality results.

Newell et al. (2017) build on (Newell et al., 2016) and predict part detection heatmaps along with real-valued tags for each pixel location. The network is trained using a novel associative embedding loss which enforces that the tags for the locations that belong to the same person are similar and dissimilar otherwise. Given a set of body joint candidates with corresponding tags, the authors demonstrated that a simple greedy matching approach could be used to correctly associate the body parts to correct individuals, while also achieving state-of-the-art results on benchmark datasets.

Nie et al. (2018a) propose an approach similar to Hough voting where each body part location votes for their centroid (torso). They also build on (Newell et al., 2016) and regress dense offset vectors from the global joint positions to the centroids of persons. Since the offset regression can be inaccurate, they first project the offsets to an embedding space where the offsets pointing towards similar locations lie close and far-apart otherwise. Finally, a greedy matching approach is employed to perform person association based on their proximity in the embedding space. Papandreou et al. (2018) also employ a similar approach, but instead of predicting displacement offsets from the centroid, they predict offsets between each pair of keypoints. Besides, they also regress short-range offsets from the locations around the body joint. The short-range offsets point towards the center of

the corresponding body joint, and are used to refine the long-range offsets iteratively. They also employ a greedy approach based on Hough-voting to group the joints belonging to the same individual.

Xia et al. (2017) propose to jointly solve multi-person pose estimation and semantic part segmentation using a single multi-task network architecture. The proposed method exploits the complementary nature of both tasks and utilizes body pose information as object-level shape prior for part segmentation, and leverages part-level segmentation to constrain the variation in pose locations. Initially, a two-branch network is used to generate part-heatmaps, part-neighbor prediction maps, and semantic part scoremaps. Afterward, all three maps are combined in a fully-connected CRF to produce final pose estimates.

2.1.3. Multi-Person Pose Estimation and Tracking

In contrast to single-person scenarios, multi-person pose estimation in videos is far-less straightforward. The existing methods for multi-person pose estimation in images cannot be applied directly to this problem since it also requires to solve the problem of person association over time in addition to the pose estimation for each person. Moreover, a working approach has to deal with large pose and scale variations, fast person or camera motions, and a varying number of persons and visible body parts due to occlusion or truncation.

While the problem has not been studied quantitatively in the literature, there exist early works towards the problem (Izadinia et al., 2012; Andriluka et al., 2008). These approaches, however, do not reason jointly about pose estimation and tracking, but instead focus on multi-person tracking alone. The methods follow a multi-staged strategy, *i.e.*, they first estimate body part locations for each person separately and subsequently leverage body part tracklets to facilitate person tracking.

For a very long time, the problem was never addressed quantitatively in the literature, mainly due to its complexity and unavailability of the annotated dataset. In Chapter 5, we will present our work which was the first to introduce and quantitatively evaluate the problem[1]. We will also present a novel approach that jointly models multi-person pose estimation and tracking in a single formulation, and also a challenging "Multi-Person PoseTrack" dataset along with a completely unconstrained evaluation protocol. In Chapter 8, we will further evolve the dataset to a large-scale benchmark termed "PoseTrack Benchmark" and provide strong baseline methods for analyzing the challenges presented by this problem.

While the problem is still in its infancy, following our work, a few recent works have also addressed the problem. Girdhar et al. (2018) propose a two-stage approach which builds on Mask R-CNN (He et al., 2017) and extends it to 3D to exploit temporal information for more robust pose estimation. The input to the network is a clip containing multiple adjacent frames taken from the video. The 2D convolutional layers in Mask R-CNN are inflated to 3D convolutions (Carreira and Zisserman, 2017b), and the region proposal network is replaced by the tube proposal network which generates tube proposals instead of bounding-box proposals. The tube proposals provide spatiotemporal locations of each person hypothesis in the input clip. The features extracted from the tube proposals are subsequently passed to the pose estimation branch which generates part heatmaps for the person hypothesis enclosed by the tube. In the second stage, the obtained poses are linked temporally using a greedy bipartite matching using Hungarian matching algorithm. A very simple similarity metric based on the IoU of person bounding boxes is used to obtain the similarity score.

[1] Contemporaneously with our work, the problem was also studied in (Insafutdinov et al., 2017).

More recently, Xiao et al. (2018) also adopt a similar approach. Given the poses estimated using a top-down approach based on ResNet (He et al., 2016) and Faster R-CNN (Ren et al., 2015), they use a greedy matching approach as in (Girdhar et al., 2018) to perform person association over time. To obtain the similarity measure between the poses in two frames, the poses from the first frame are propagated to the second frame using optical flow information, and Object Keypoint Similarity (OKS) (Ronchi and Perona, 2017) is used to measure the similarity. Although the approach relies on a simple tracking algorithm, it reports very good results, mainly due to the powerful pose estimation model.

2.2. 3D POSE ESTIMATION

The estimation of 3D pose from RGB images increases the challenges even further since the depth of the joints also has to be estimated. It is an inherently ill-posed problem due to the missing depth and scale ambiguities present in monocular images. The problem has been studied for over three decades (Lee et al., 1985; Rehg and Kanade, 1994; Heap and Hogg, 1996), but it is still considered extremely challenging. Therefore, a significant fraction of methods focuses on images with only one, pre-localized, person. A few recent works have also addressed it for multiple interacting people. In this section, we will first provide an overview of the approaches for single-person 3D pose estimation followed by a review of the approaches that also tackle multiple persons. As in the last section, we will discuss methods for 3D body pose or 3D hand pose together without any explicit differentiation, until stated otherwise.

2.2.1. Model-based methods

The model-based methods represent the articulated 3D pose using a 3D shape model. Earlier methods used primitive shapes such as cylinders (Sidenbladh et al., 2000), ellipsoids (Sminchisescu and Triggs, 2001) or Gaussian blobs (Plankers and Fua, 2001) to represent the body pose, while more recent methods use sophisticated parametric shape models such as SCAPE (Anguelov et al., 2005), SMPL (Loper et al., 2015) or MANO (Romero et al., 2017) to also capture the shape deformations of the human body or hands. This is often formulated as an optimization problem, whose objective is to find the model's deformation parameters such that its projection is in correspondence with the observed image data. Earlier methods in this direction rely on low-level visual cues such as edges (Heap and Hogg, 1996), silhouettes (Wu et al., 2001; Sigal et al., 2008), shading information (de La Gorce et al., 2011) or combinations of these (Lu et al., 2003; Guan et al., 2009). Sigal et al. (2008) are one the first to use a high-quality shape model. In particular, they use the SCAPE model proposed by Anguelov et al. (2005). The parameters of the SCAPE model are estimated by fitting it to the ground-truth silhouettes of the body. Guan et al. (2009) build on (Sigal et al., 2008) and utilize additional information such as edges and shading cues during the fitting process.

More recent approaches use automatically detected keypoint locations and use similar optimization methods to minimize the discrepancies between the detected keypoints and the projection of the corresponding keypoints from the shape models (Bogo et al., 2016; Panteleris et al., 2018). Bogo et al. (2016) used the SMPL (Loper et al., 2015) body model, which unlike SCAPE, has explicit 3D joints. They demonstrated that fitting the SMPL model using only the 2D joint locations is sufficient to obtain good shape estimates. Lassner et al. (2017) improved the approach of Bogo et al. (2016) by

introducing an extra fitting objective, denser 2D keypoint locations, and additional training data.

Tung et al. (2017) adopt a learning-based approach. Given an input image, they directly predict the deformation parameters of the 3D model using a deep neural network. The network is trained using a combination of strong supervision from synthetic data and weak supervision using 2D joint locations and silhouettes. Pavlakos et al. (2018b) also propose a similar approach but train the network with weak-supervision only, and do not require additional synthetic training data. Kanazawa et al. (2018) further extend the network architecture by adding a discriminator network that predicts whether the generated body parameters lead to a realistic body pose or not. This provides additional regularization to the generator network and prevents it from generating implausible parameters which often is the case due to re-projection ambiguities. Instead of predicting the parameters of the parametric body model, Varol et al. (2018) directly regress the volumetric shape of the human body, where the volumetric shape is represented as a 3D voxel-based occupancy grid.

2.2.2. Search-based methods

These methods follow a non-parametric approach and formulate 3D pose estimation as a nearest neighbor search problem from large databases of 3D poses. The matching is performed based on some low (Athitsos and Sclaroff, 2003; Romero et al., 2010) or high (Yasin et al., 2013; Pons-Moll et al., 2014; Chen and Ramanan, 2017) level features extracted from the image.

The approaches in (Athitsos and Sclaroff, 2003; Romero et al., 2010) use low-level image features such as edges, skin color segmentation or HoG features to search nearest synthetic images in the database, where each synthetic image contains a corresponding 3D pose. Yasin et al. (2013) use ground-truth 2D pose in the first frame and track it in the video. For 3D pose estimation at each frame, the nearest neighbor search is performed to obtain the nearest 3D pose based on the features extracted from 2D pose coordinates. Chen and Ramanan (2017) also follow a similar approach and propose a non-parametric nearest neighbor model to retrieve 3D exemplars that minimize the re-projection error from the estimated 2D joint locations. Pons-Moll et al. (2014) instead utilize high-level information about the geometric relationships, called "posebits", between body joints to retrieve semantically similar poses from a large corpus of 3D poses. The posebits are obtained using discriminatively trained binary classifiers that predict if a specific geometric relation holds in the given image or not.

2.2.3. From 2D pose to 3D

Earlier methods in this direction learn probabilistic 3D pose models from MoCap data and recover 3D pose by lifting the 2D keypoints (Ramakrishna et al., 2012; Simo-Serra et al., 2013). Ramakrishna et al. (2012) construct a sparse representation of 3D body pose using a MoCap dataset and fits it to manually annotated 2D joint positions. While Wang et al. (2014a) extend the approach to handle estimated poses from an off-the-shelf 2D pose estimator (Yang and Ramanan, 2011), Du et al. (2016) extend it to leverage temporal information in video data. Simo-Serra et al. (2012, 2013) use the information about the 2D body joints to constrain the search space of 3D poses. In (Simo-Serra et al., 2012) they propose an evolutionary algorithm to sample poses from the 3D pose space that correspond to the estimated 2D joint positions. This set is then exhaustively evaluated according to some anthropometric constraints. The approach is extended in (Simo-Serra et al., 2013) such that the 2D pose estimation and 3D pose estimation are iterated. In contrast to (Ramakrishna et al., 2012; Wang et al., 2014a; Simo-Serra et al., 2012), the approach (Simo-Serra et al., 2013) also deals with

2D pose estimation errors. Tome et al. (2017) propose a probabilistic 3D pose model and combine it with a multi-staged CNN, where the CNN incorporates evidence from the 2D body part locations and projected 3D poses to sequentially improve 2D joint predictions which in turn also results in better 3D pose estimates. Action-specific priors learned from motion capture data have also been proposed for 3D pose tracking (Urtasun et al., 2006). These approaches, however, are more constrained by assuming that the type of motion is known in advance and therefore cannot deal with a large and diverse pose dataset.

Other approaches adopt a data-driven approach and use deep neural networks to learn a direct mapping from 2D pose to 3D (Moreno-Noguer, 2017; Martinez et al., 2017; Zimmermann and Brox, 2017). Martinez et al. (2017) propose a deep neural network with residual connections to directly regress 3D pose from 2D pose as input. Moreno-Noguer (2017), on the other hand, propose first to encode 3D pose using an Euclidean distance matrix formulation that implicitly incorporates body joint relations and allows to regress 3D poses in the form of a distance matrix. Instead of 2D keypoint locations, Zimmermann and Brox (2017) and Tekin et al. (2017) use 2D heatmaps (Wei et al., 2016; Newell et al., 2016) as input and learn convolutional neural networks for 3D pose regression. The approach in (Zimmermann and Brox, 2017) is one of the first learning-based methods to estimate 3D hand pose from a single RGB image. They use an existing 2D pose estimation model (Wei et al., 2016) to first obtain the heatmaps of hand keypoints and feed them to another CNN that regresses a canonical pose representation and the camera view point.

More recently, Pavlakos et al. (2018a) and Shi et al. (2018) build on the idea of "posebits" and combine them with 2D pose information to directly regress 3D body pose using a deep neural network. Both approaches demonstrate that "posebits" like geometric relationships provide useful information about the 3D pose, and helped to achieve state-of-the-art results on benchmark datasets. The main benefit of "posebits" is that, unlike 3D pose annotations, they can be annotated reasonably easily for any image taken from the wild. Hence, data-driven robust classifiers can be learned to predict these relationships from unconstrained images.

The aforementioned methods have the advantage that they do not necessarily require images with ground-truth 3D pose annotations for training, their major drawback is that they cannot handle reprojection ambiguities (a joint with positive or negative depth will have the same 2D projections). Moreover, they are sensitive to errors in 2D image measurements and the required optimization methods are often prone to local minima due to incorrect initializations.

2.2.4. 3D pose from images

These approaches aim to learn a direct mapping from RGB images to 3D pose. Earlier approaches in this direction utilize discriminative methods to learn a mapping from hand-crafted local image features (e.g., HOG, SIFT, etc.) to 3D human pose (Bo et al., 2008; Mori and Malik, 2006; Bo and Sminchisescu, 2010; Agarwal and Triggs, 2004; Sminchisescu et al., 2005; Agarwal and Triggs, 2006). Since local features are sensitive to noise, Kostrikov and Gall (2014) proposed an approach based on a 3D pictorial structure model that combines generative and discriminative methods to obtain robustness to noise. For this, regression forests are trained to estimate the probabilities of 3D joint locations and the final 3D pose is inferred by the pictorial structure model. Since inference is performed in 3D, the bounding volume of the 3D pose space needs to be known, and the inference requires a few minutes per frame. In addition to the local image features, (Ionescu et al., 2014a)

also utilize body part segmentation with a second-order hierarchical pooling process to obtain robust image descriptors. Tekin et al. (2016b) collect image evidence not only from a single frame, but a set of neighboring frames in a fixed temporal window. The approach starts by detecting person's bounding boxes in the first frame, and simultaneously tracks and aligns the bounding boxes in the remaining frames. For a given test frame, a set of neighboring frames appearing in a temporal window are used to extract 3-dimensional HoG features. These features are then fed into a discriminative model to directly regress the 3D pose of the person in the test frame. Lin et al. (2017a), on the other hand, propose to use a recurrent neural network to enforce temporal consistency in video sequences where the pose at each frame is estimated while also utilizing the predictions from previous frames.

More recent approaches learn end-to-end CNNs to regress the 3D joint locations directly from the image. The work of Li and Chan (2014) is one of the earliest methods that presents an end-to-end CNN architecture. A multi-task loss is proposed to simultaneously detect body parts in 2D images and regress their locations in 3D space. In (Li et al., 2015), a max-margin loss is incorporated with a CNN architecture to model joint dependencies efficiently. Similarly, Zhou et al. (2016b) enforce kinematic constraints by introducing a differentiable kinematic function that can be combined with a CNN. The approach of Tekin et al. (2016a) uses auto-encoders to incorporate dependencies between body joints and combines them with a CNN architecture to regress 3D poses. Sun et al. (2017b) propose a bone-based pose representation and a compositional loss that encodes long-range dependencies between body parts and allows efficient 3D pose regression. Zhou et al. (2016a) present an expectation maximization algorithm to estimate 3D poses from monocular videos, where additional smoothness constraints are used to exploit the temporal information in videos.

While these methods can better handle 2D projection ambiguities, their main downside is that they are prone to over-fitting to the views only present in training data. Thus, they require a significant amount of training data with accurate 3D pose annotations. Collecting massive amounts of training data in unconstrained environments is, however, infeasible. To this end, approaches for data augmentation have also been proposed. The methods generate synthetic images with high realism and accurate ground-truth 3D poses to enlarge the training data (Rogez and Schmid, 2016; Chen et al., 2016; Ghezelghieh et al., 2016; Varol et al., 2017; Mueller et al., 2017).

Other approaches formulate the problem in a multi-task setup to jointly estimate both 2D keypoint locations and 3D pose (Popa et al., 2017; Pavlakos et al., 2017; Sun et al., 2017b; Zhou et al., 2017b; Nie et al., 2017; Luvizon et al., 2018; Yang et al., 2018). This allows transferring knowledge from unconstrained images with only 2D keypoint annotations to 3D pose estimation models trained using constrained 3D pose annotations. While Park et al. (2016) directly use the 2D joint coordinates to regularize the training of a CNN, Tekin et al. (2017) and Popa et al. (2017) use confidence scoremaps of 2D body joints obtained using a CNN as additional features for 3D pose regression.

2.2.5. Multi-Person 3D Human Pose Estimation

All of the approaches for 3D human pose estimation discussed so far assume that only a single person is visible. Similar to multi-person pose estimation in 2D, multi-person pose estimation in 3D also has to deal with the significant amount of occlusion and truncation, different scale of persons, and a variable number of persons in the image. In addition to all these challenges, the depths of the joints also have to be estimated.

The approach of Rogez et al. (2017) was one of the first in this direction. They build on a similar

idea as Faster-RCNN or Mask R-CNN (Ren et al., 2015; He et al., 2017). First, a region proposal network is used to generate a set of candidate locations of the persons. Each proposal is then classified either as background or as one of the pose types. The pose types are obtained by clustering a large set of poses and using the center of each cluster as anchor pose. Given the person bounding box and anchor pose type, the final 3D pose for each proposal is estimated by using a regression branch tailored for a specific pose type. The main contribution of their approach toward multi-person 3D pose estimation is to perform person localization and pose estimation in a single pipeline.

Similar in idea, Mehta et al. (2017b) build upon (Cao et al., 2017). In addition to part heatmaps and part-affinity maps, they also produce location maps (Mehta et al., 2017a) and occlusion-robust pose maps (ORPM). The location maps are 3-channel feature channels where each channel contains the X, Y, or Z coordinate of a joint at the location of its 2D projection in the image, while occlusion-robust pose maps store the 3D location of each joint not only at its 2D pixel location but also at several other locations decided by the limbs of the corresponding person. This redundancy introduced in ORPM allows estimating the pose even if persons are substantially occluded. During inference, the joint-to-person association is performed as in (Cao et al., 2017) and the 3D joint locations are taken either from location maps or occlusion-robust pose maps.

It is important to note that both of these approaches, like most of the methods for single-person pose estimation, estimate root-relative 3D pose. This means that the absolute depth of the persons with respect to the camera is not known, *i.e.*, it is not known which person is closer to the camera or which person is farther away. In this way, both methods do not bring any additional information as compared to a simple approach where one can first detect the persons using a person detector followed by any of the single-person 3D pose estimation methods discussed in the previous section. To obtain the absolute depth of the persons, their relative sizes and position with respect to the camera have to be estimated, both of which are very challenging to estimate due to projection ambiguities, *i.e.*, a small person close to the camera will have a similar size in 2D projection as compared to a tall person standing farther away from the camera.

More recently, Zanfir et al. (2018) propose a framework to estimate 2D and 3D pose, shape, and position of all persons in a single framework. First, the 2D pose, root-relative 3D pose and part segmentation of each person are estimated using an off-the-shelf approach (Popa et al., 2017). Subsequently, the parameters of the shape and position are optimized by fitting SMPL (Loper et al., 2015) to the 2D projections. Given the initial estimates for each person, a joint objective function is proposed that refines the initial estimates by penalizing simultaneous 3D volume occupancy between different persons in the scene, and incorporating the constraint that some of the persons in the scene share a common supporting plane. In case of videos, the obtained poses are tracked using a simple Hungarian matching approach, and all estimates are once again refined by optimizing the objective function over time while also enforcing smoothness constraints, such as by imposing constant velocity of pose angles over time. The approach demonstrates very good results under occlusions, different positon and close interactions to some degrees. However, the scale ambiguities are still not addressed.

CHAPTER 3

Preliminaries

In this chapter, we provide a brief review of the primary methods and tools that are used in the later chapters. We briefly introduce the basic building blocks of convolutional neural networks and describe the motivation behind the design choices made by the state-of-the-art network architectures for single-person human pose estimation. We review the problem of graph partitioning and node labeling and describe the branch-and-cut method to solve integer linear programs. We also describe the evaluation metrics that are used to quantitatively evaluate the developed methods in this thesis.

Contents

3.1. CONVOLUTIONAL NEURAL NETWORKS

In this section, we will briefly describe the fundamentals of convolutional neural networks (CNN) (LeCun et al., 1998), and rationales behind the design choices for the commonly used CNN architectures for human pose estimation.

3.1.1. Building Blocks of Convolutional Networks

An example of a CNN architecture can be seen in Figure 3.1. In general, a CNN consists of sequentially arranged convolutional layers where each convolutional layer is often followed by a pooling layer. A convolutional layer consists of a set of linear convolutional kernels and an element-wise non-linear activation function.

Let $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$ be an RGB image provided as an input to a CNN with L layers, where h and w correspond to the height and width of the image, respectively. Each layer in the CNN takes as input

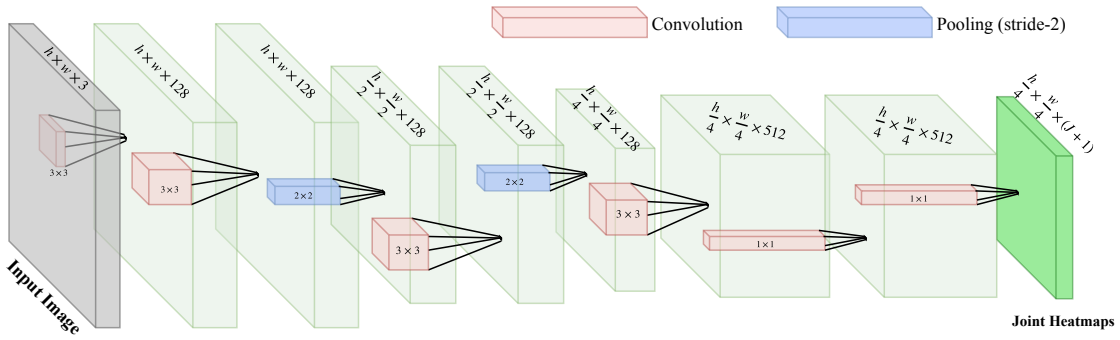


Figure 3.1: Example of a convolutional architecture. A CNN architecture consists of a set of sequentially arranged convolutional layers which are often followed by a pooling layer. In this figure a fully-convolutional network (Long et al., 2015) is shown which produces a set of feature maps as output, where each output feature map provides dense pixel-wise predictions. For pose estimation, the output feature maps generally correspond to the likelihood of the presence of body joints at each pixel location, and are often referred to as "Heatmaps".

a set $\mathbf{G}_{l-1} \in \mathbb{R}^{h' \times w' \times C_{l-1}}$ of C_{l-1} feature maps produced by the previous layer $l-1$, and produces a set $\mathbf{G}_l \in \mathbb{R}^{h'' \times w'' \times C_l}$ of C_l feature maps as output. The input to the first layer $l = 1$ is the input image *i.e.*, $\mathbf{G}_0 = \mathbf{I}$.

3.1.1.1. Convolutional Layer

The convolutional layer consists of C_l linear convolutional kernels, each with a weight matrix $W_l^c \in \mathbb{R}^{k \times k \times C_{l-1}}$ and a bias value b_l^c, where k is the size of the kernel. The c_{th} channel $G_l^c \in \mathbf{G}_l$ in the output feature maps of the convolutional layer corresponds to the response of convolving the c_{th} filter over the input feature maps:

$$G_l^c = \sigma(\mathbf{G}_{l-1} * W_l^c + b_l^c), \tag{3.1}$$

where σ is a non-linear activation function that is applied element-wise to each location. There exist several non-linear activation functions in the literature such as logistic sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU) (Nair and Hinton, 2010). However, ReLU is the most popular activation function since it yields better convergence for the deep neural networks and is much faster to compute as compared to other options (LeCun et al., 2015). The ReLU is defined as,

$$\sigma(g) = \max(g, 0). \tag{3.2}$$

3.1.1.2. Pooling Layer

The pooling layer replaces the activation values at each feature map location with the summary statistic from a local neighborhood. The most commonly used pooling operations are the average-pooling or max-pooling. As evident from the names, average-pooling replaces the activation values with the average of the local neighborhood and max-pooling replaces the values with the maximum value from the local neighborhood. The pooling is often applied to non-overlapping regions of the feature map which results in down-sampled feature maps as output. The pooling operation has the benefit that it provides invariance to small translations, and the resulting down-sampled feature maps

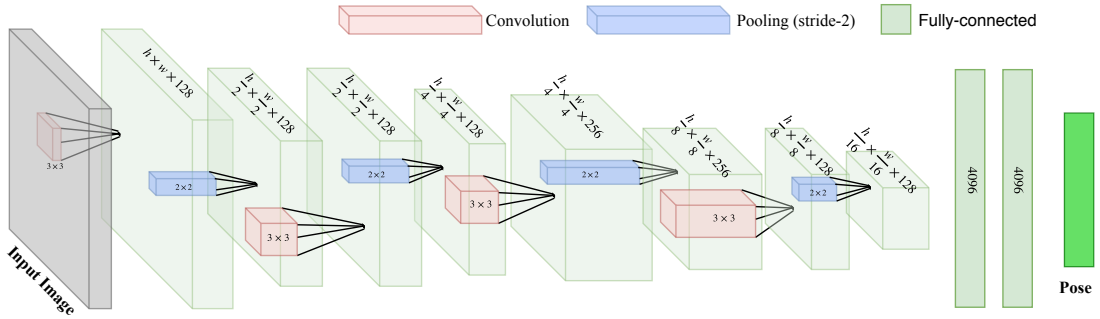


Figure 3.2: A CNN architecture with fully-connected layers at the end. Every convolutional layer is followed by a pooling layer to down-sample the feature maps to a reasonably small resolution before applying the fully-connected layers. For pose estimation, the last layer usually corresponds to the coordinates of the body joints.

improve the computational efficiency of the network since the subsequent layers have increasingly fewer inputs to process (see Figure 3.1 and 3.2). Further, the down-sampling operation increases the effective receptive field of the network which helps the network to learn high-level semantic information in the later stages of the network.

3.1.1.3. Fully-Connected Layer

The fully-connected (FC) layers are sometimes added at the end of CNNs to map the input image to a fixed number of output values (*e.g.*, class labels). In contrast to a convolutional layer which computes responses only from a local neighborhood, every output response in an FC layer is computed by using the values from all input feature map locations. An example of a CNN with FC layers can be seen in Figure 3.2. The use of pooling layers is, in particular, important for networks with FC layers, since the number of weights in an FC layer increases exponentially with the size of input feature maps. Therefore, generally, several pooling layers are introduced in the networks with FC layers to down-sample the feature maps to a reasonable size before applying the FC layer.

3.1.1.4. Loss Function

In order to train the network, a loss function $\mathcal{L}(\mathbf{p}, \hat{\mathbf{p}})$ is defined which measures the deviation between the network's output $\mathbf{p}$ for any training sample and the desired ground-truth value $\hat{\mathbf{p}}$. The commonly adopted loss functions for human pose estimation include L_1 or L_2 norm and sigmoid with cross-entropy. The norms are used when pose estimation is casted as a regression problem, *i.e.*, regression of pose coordinates using a fully-connected network (see Figure 3.2) or regression of predefined joint heatmaps using a fully-convolutional network (Long et al., 2015) (see Figure 3.1). The sigmoid with cross-entropy loss function is used when pose estimation is treated as a classification task, *i.e.*, classification of each pixel to a particular joint type or the background class.

3.1.1.5. Training

Given the loss function, the network is trained by seeking for the network's parameter that minimize the empirical risk over a provided training data $\mathcal{T}$, given as,

$$E(\mathbf{W}, \mathcal{T}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(\mathbf{p}_n, \hat{\mathbf{p}}_n), \tag{3.3}$$

where N is the total number of training samples in the training data, and $\mathbf{W}$ is the vector containing all the learnable parameters in all the layers of the network. The minimization of (3.3) is done using an optimization procedure *i.e.*, by performing gradient descent on the total loss E as,

$$\mathbf{W}_t \leftarrow \mathbf{W}_{t-1} + \alpha \nabla E(\mathbf{W}, \mathcal{T}), \tag{3.4}$$

where α is a learning rate and $\nabla E(\mathbf{W}, \mathcal{T})$ are the gradients of the parameters $\mathbf{W}$ computed using the back-propagation algorithm (Rumelhart et al., 1986; LeCun et al., 1998). In practice, the mini-batch gradient descent method is used which instead uses a small batch of randomly chosen training samples to perform the update step (3.4). We refer the readers to (LeCun et al., 2015; Goodfellow et al., 2016) for further details.

3.1.2. Convolutional Neural Networks for Human Pose Estimation

The convolutional neural networks have achieved breakthrough performance on many tasks in computer vision, including single-person human pose estimation (Wei et al., 2016; Newell et al., 2016). However, naively combing the building blocks discussed above can result in sub-optimal performance. The state-of-the-art CNN architectures for single-person human pose estimation are, therefore, carefully designed and tailored for the problem in hand.

The main complexities in estimating articulated human pose from an image arise due to large degrees of freedom of the human body, and the vast amount of appearance variation due to varied body articulations, clothing and imaging condition. In particular, the small body parts such as wrists and elbows exhibit a large amount of appearance variation. To this end, it has been a common practice to exploit the spatial context of the human body, *i.e.*, the location of one body part can provide strong cues about the location of other parts. Earlier methods for human pose estimation incorporated such cues by utilizing graphical models that model the interdependencies between body parts (Felzenszwalb and Huttenlocher, 2005). On the other hand, the state-of-the-art methods (Wei et al., 2016; Newell et al., 2016) use fully-convolutional neural networks (Long et al., 2015) and design the network architecture such that it can implicitly exploit such spatial contextual information. The two main factors that are considered by these methods are a multi-staged network design and a large receptive field of the network.

3.1.2.1. Multi-staged Design

Some of the parts in the human body such as the face or neck can be detected easily as compared to the shoulders or elbows. Further, the location of the face can provide strong cues about the shoulders. This is, precisely, the rationale behind the multi-staged network designs. The first stage of the networks utilizes only the image evidence to produce the body part locations. At this point, some of

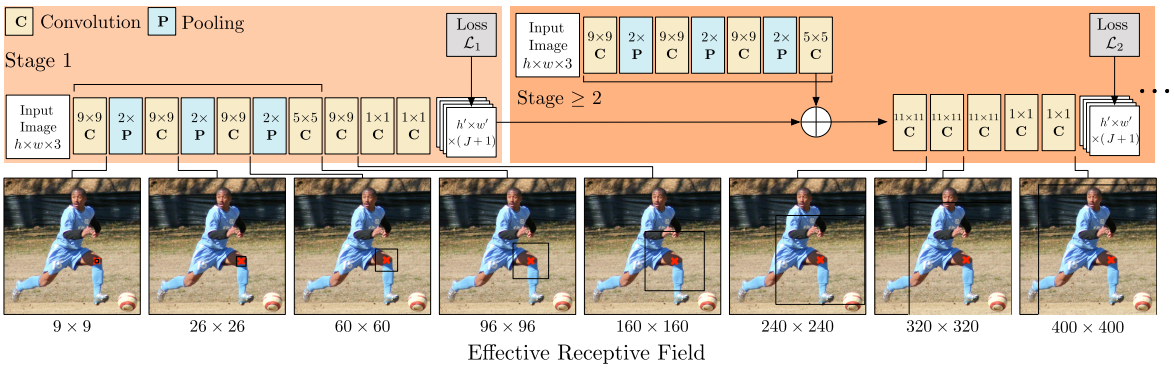


Figure 3.3: The multi-staged network architecture proposed by Wei et al. (2016). The first stage utilizes only the image evidence while all subsequent stages also utilize the body part predictions from the preceding stages. The black bounding boxes represent the effective receptive field of the network at the given layer. Local supervision is provided to each layer by enforcing the loss $\mathcal{L}$ at the output of each stage. The figure is adopted from (Wei et al., 2016).

the body parts may be mislocalized, however, the locations of the easy-to-detect body parts can be used to refine the locations of other body parts. Hence, the subsequent stages in multi-staged network architectures also use the predictions from the preceding stages as input. This allows the networks to exploit the dependencies between body parts and to improve the part detection performance sequentially.

3.1.2.2. Large Receptive Field

Even naively creating a multi-staged network architecture can result in sub-optimal performance. For example, a network with many stages but with 1×1 convolutional layers will never be able to exploit the spatial context due to its limited receptive field. Therefore, designing the network to have larger receptive fields is also crucial for better performance. The large receptive fields can be obtained by introducing pooling layers, by increasing the kernel size of the convolutional filters or by increasing the depth of the network. However, each of these comes with its trade-off. The pooling layers result in reduced spatial resolution, hence, reduce the network's precision to localize the body parts accurately. Increasing the kernel size of the convolutional filters increases the number of parameters, and the increased depth can cause vanishing gradients during training.

In general, a down-sampling factor of 8 is known to preserve most of the information to perform per pixel predictions accurately (Long et al., 2015; Yu et al., 2017). Therefore, the pooling layers should be introduced such that the down-sampling ratio is not more than 8. Further, it is preferred to use multiple convolutional layers with smaller kernel sizes than to use one convolutional layer with a large kernel size. The rationale behind this is that the effective receptive field of two consecutive convolutional layers with a kernel size of 3×3 is the same as the receptive field of a single layer with a kernel size of 5×5, but it results in fewer network parameters $(3 \times 3 + 3 \times 3)$ vs. (5×5). Further, the stacked convolutional layers also incorporate two non-linearities which result in an increased representational power of the network (Simonyan and Zisserman, 2014). The use of stacked convolutional layers, however, can result in very large depths of the network. To this end, Wei et al. (2016) proposed to use local supervision for each stage to avoid vanishing gradients. An overview

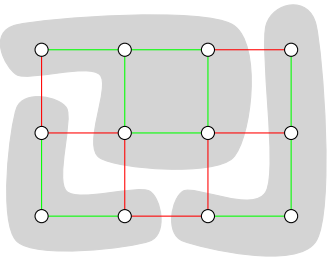


Figure 3.4: An example of a partitioning of a graph into three subsets of nodes (grey). The green color indicates that the edge between two nodes is not-cut, while red color indicates that the edge is cut.

of the multi-staged network architecture proposed by Wei et al. (2016) can be seen in Figure 3.3. They combined all of the strategies mentioned above and developed a network architecture with an effective receptive field of 400×400 at the end of the second stage. Depending on the scale of the human body, such a large receptive field is sufficient to exploit the context of the complete human body.

3.2. GRAPH PARTITIONING

Graph partitioning is one of the most fundamental problems in computer vision. In fact, any image or video can be represented as a graph with nodes of the graph being the pixels in an image or frames in a video. A partitioning of a graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ is then the partition of its nodes into k distinct sets of nodes such that each set $\mathcal{D}_i \subset \mathcal{D}$, $\mathcal{D} = \bigcup_i^k \mathcal{D}_i \; \forall i \in \{1, \cdots, k\}$, and $\mathcal{D}_i \cap \mathcal{D}_j = \varnothing \; \forall i \neq j$. An example of such partitioning can be seen in Figure 3.4. The partitioning of a graph is, particularly, of great importance for problems where no information about the number or size of the partitions is known, and only the measure of similarity or dissimilarity for each pair of nodes is available. Graph partitioning is also referred to as graph decomposition, correlation clustering or minimum cost multicut problem in the literature (Chopra and Rao, 1993).

Many problems in computer vision, for example image decomposition (Arbelaez et al., 2011; Keuper et al., 2015) and multi-target tracking (Tang et al., 2016) can be viewed as problems of graph partitioning. For example, in case of image decomposition, the nodes of the graph consist of all pixels in the image, and the decisions to be taken are whether two pixels (nodes) belong to the same object in the image or not. If the pixels belong to the same object, the edge between both pixels should not be cut and should be cut otherwise. For multi-target tracking, the nodes of the graph consist of the locations of all possible targets over a time span, and the goal is to keep the edges that associate the same targets and cut the edges between different targets.

Formally, given a graph G we want to find the set of edges $M \subset \mathcal{E}$ that are cut and partition the graph into connected subsets of nodes. Such edges are shown in red in Fig 3.4, and are called as the "multicut" of the graph G. Or inversely, we want to find the set of edges $N \subset \mathcal{E}$ that are not-cut and group the nodes into distinct connected components. Such edges are shown in green in Fig 3.4. While both formulations are valid, *i.e.*, $M \cap N = \varnothing$, in this thesis we follow the latter. Given, for every edge $(d, d') \in \mathcal{E}$ connecting the nodes $d \in \mathcal{D}$ and $d' \in \mathcal{D}$, a cost $\psi(d, d') \in \mathbb{R}$ that is negative when both nodes should be connected and positive otherwise, the partitioning problem can be written

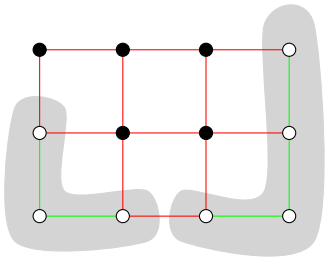


Figure 3.5: An example of graph partitioning and node labeling. The nodes are either labeled 1 (white) or 0 (black). The nodes with label 1 are partitioned into two subsets (grey). The green color indicates that the edge between two nodes is not-cut, while red color indicates that the edge is cut. Note that no no-cut edge exists between the pair of nodes where at least one of the nodes is labeled 0.

as the optimization of the following objective

$$\min_{N \in \mathcal{N}_G} \sum_{(d,d') \in N} \psi(d, d'), \tag{3.5}$$

where $\mathcal{N}_G$ is the set of all possible configurations of not-cut edges. The rationale behind this formulation is that grouping the nodes that belong to the same instance should cost less as compared to grouping the nodes that belong to different instances. The problem can be further reformulated as 01-labeling of the edges using a constrained binary Linear Program (LP) with a binary random variable $s \in \{0, 1\}^{|\mathcal{E}|}$, where $s_{(d,d')} = 1$ indicates that the edge $(d, d') \in \mathcal{E}$ connecting the nodes d and d' is labeled 1 or in other words is not-cut:

$$\arg\min_{s} \sum_{(d,d') \in \mathcal{E}} s_{(d,d')} \psi(d, d'), \tag{3.6}$$

subject to

$$s_{(d,d')} \leqslant \sum_{(d'',d''') \in C \setminus \{(d,d')\}} s_{(d'',d''')}, \quad \forall C \in \text{cycles}(\mathcal{G}). \tag{3.7}$$

The cyclic inequalities in (3.7) ensure that every subset $\mathcal{D}_i$ is connected. The objective (3.6) seeks for the configuration of edges such that the overall cost is minimized. In general, we may require additional constraints to obtain feasible solutions depending on the problem in hand.

Optimizing the objective (3.6) is an NP hard problem (Bansal et al., 2004) and can be solved using Integer Linear Programming (ILP) with additional constraints to enforce that the solutions are binary, *i.e.*, $s_{(d,d')} \leqslant 1, \forall (d, d') \in \mathcal{E}$. There exist a large number of methods for solving ILP problems. In this thesis, we use the commonly adopted branch-and-cut method. We will briefly explain it in Section 3.2.2.

3.2.1. Graph Partitioning and Node Labeling

In the above, we discussed the problems that only require partitioning the graph $\mathcal{G}$ into disjoint connected sets of nodes. However, many problems also require labeling of the nodes. For example,

object detection or instance segmentation both come into this category, where in addition to identifying distinct instances we may also need to label them with a class label. In this section, we will discuss the problems that only require binary labeling of the nodes. However, formulating the problems that may need to choose from more than two labels is also possible (Kroeger et al., 2014; Pishchulin et al., 2016; Levinkov et al., 2017; Insafutdinov et al., 2016).

A simple example of graph partitioning and binary node labeling is person segmentation where the goal is to segment all visible persons in the image from the background, and identify different instances. In this case, for every pixel we need to decide if it depicts a person, and for every pair of pixels that depict a person, we need to decide if they belong to the same person.

Formally, given a graph $G = (\mathcal{D}, \mathcal{E})$, we need to assign a binary label to all nodes, and partition the graph such that all nodes with label 1 are grouped into distinct components. An example of such graph partitioning and node labeling can be seen in Figure 3.5. The problem can be written as constrained binary LP with two random variables $v \in \{0,1\}^{|D|}$ and $s \in \{0,1\}^{|\mathcal{E}|}$, where $v_d = 1$ indicates that the node is labeled 1 and $(v_d = 0)$ indicates the node is labeled 0. For every pair of nodes (d, d'), the edge $s_{(d,d')}$ is constrained such that $s_{(d,d')}$ can be 1 only if both nodes are labeled 1, *i.e.*, $v_d = 1$ and $v_{d'} = 1$. Given, for every node a cost ϕ_d that is negative if the node should be labeled 1 $(v_d = 1)$, and positive otherwise $(v_d = 0)$, and for every edge (d, d') a cost that is negative when both nodes should be connected $(s_{(d,d')} = 1)$ and positive otherwise $(s_{(d,d')} = 0)$, the graph partitioning and node labeling problem can be written as the optimization of the following objective

$$\underset{(v,s)}{\arg\min} \sum_{v_d \in \mathcal{D}} v_d \phi(d) + \sum_{(d,d') \in \mathcal{E}} s_{(d,d')} \psi(d, d') \tag{3.8}$$

subject to,

$$s_{(d,d')} \leqslant \sum_{(d'',d''') \in C \setminus \{(d,d')\}} s_{(d'',d''')}, \quad \forall C \in \text{cycles}(\mathcal{G}), \tag{3.9}$$

$$s_{(d,d')} \leqslant v_d \wedge s_{(d,d')} \leqslant v_{d'} \quad \forall (d, d') \in \mathcal{E}. \tag{3.10}$$

Same as before, the cyclic inequalities in (3.9) ensure that every subset $\mathcal{D}_i \in \mathcal{D}$ is connected. The constraints (3.10) ensure that an edge between two nodes is added only when both nodes are labeled 1 (see Figure 3.5). The problem can be solved using ILP with a branch-and-cut method as described next.

3.2.2. Branch-and-Cut Method

The branch-and-cut (Padberg and Rinaldi, 1991) method follows a branch-and-bound algorithm that systematically explores the candidate solutions in form of a rooted tree. It maintains a binary tree with the original ILP problem at the root node and splits the problem at each node into two sub-problems (children) by means of a branching algorithm such that each sub-problem has a narrowed feasible region than the ILP problem at the current node. For this, at each node, it solves the LP relaxation of the problem (*i.e.*, ILP without the integrality constraints) using the simplex method which may provide non-integer solutions. These non-integer solutions are then used to devise two constraints for a chosen branching variable that are satisfied by all feasible integer solutions of the problem but violated by the current non-integer solution. Each of the two constraints is added separately to the current ILP problem, and the two subproblems are created. This process is called "branching". There exist several methods to choose the branching variable. The simplest strategy is "Most Infeasible

Algorithm 1 Branch-and-Cut method to solve Integer Linear Programs

1: Add the initial ILP to T, the list of problems that are not branched
2: Set $s^* = \text{null}$ and $v^* = -\infty$
3: While T is not empty
   a. Select and remove a problem from T
   b. Solve the LP relaxation of the problem using simplex method.
   c. If the solution is infeasible, go back to 3.
   d. Denote the solution by s with objective value v.
   e. If $v \geqslant v^*$, go back to 3.
   f. If s is integer, set $v^* \leftarrow v$, $s^* \leftarrow s$, and go back to 3
   g. Branch the problem into two sub-problems with narrowed feasible regions.
   h. Add these problems to T and go back to 3
4: return s^*

Branching" that chooses the variable whose non-integer solution ${s'}_e$ has the highest fractional value (*i.e.*, closer to 0.5). The two constraints then take the form $s_i \leqslant \lfloor {s'}_i \rfloor$ and $s_i \geqslant \lceil {s'}_i \rceil$ and. The LP relaxations for both sub-problems are also solved independently using the simplex method. If the solution of any of the sub-problems is infeasible, the corresponding node is deleted from the tree. Subsequently, the solutions at all the ending-nodes of the tree are compared, and the node with the lowest objective value v^* is then chosen for further branching. This process is repeated until an optimal integer solution is found. In general, the optimal solution is observed when a feasible integer solution is found at a node and the objective at that node is smaller than or equal to the objectives at any other ending-node of the tree. The whole process can be implemented by maintaining a list of nodes that have not been branched (*i.e.*, ending-nodes) as shown in the pseudo-code provided in Algorithm 1.

3.3. EVALUATION METRICS

In the following we will briefly discuss the commonly used evaluation metrics for human pose estimation and multi-target tracking.

3.3.1. 2D Pose Estimation

3.3.1.1. Percentage of Correct Parts (PCP)

PCP, first proposed by Ferrari et al. (2008), considers a body part (limb) to be correctly localized if both of its endpoints are within a certain threshold from the ground-truth locations. The threshold is defined adaptively for each part as the 50% of its ground-truth length. PCP is then computed as the percentage of correctly localized body parts. The main limitation of PCP is that it results in a very strict threshold for foreshortened body parts since the pixel-distance between both endpoints under forshortening is generally very small. Thus, the foreshortened parts have to localized with very high precision to achieve a higher PCP value.

3.3.1.2. Percentage of Correct Keypoints (PCK)

PCK, first proposed by Sapp et al. (2011), instead considers each body joint independently. A body joint is considered correctly localized if its predicted location falls within a certain threshold from the ground-truth location. There exist many variants to calculate the threshold values. Yang and Ramanan (2013) consider the height (maximum side of bounding box) of the person to calculate the threshold. The drawback of this definition is that it is dependent on the body articulation, *i.e.*, the threshold value for an up-right person is a lot higher than a person who is bowing down. Sapp et al. (2011) instead proposed to use the height of the torso to make the thresholds less dependent on body articulation. However, it still has the limitation that the threshold becomes very strict in case of a foreshortened torso. To this end, Andriluka et al. (2014) proposed to use the length of the head segment to calculate the threshold for all body joints. This makes the PCK metric completely independent of body articulation. This variant of PCK is referred to as "PCKh". PCKh is the most commonly used evaluation measure for 2D pose estimation. While 50% of the head-segment length is the commonly used thresold value, some works also plot the accuracy as the function of whole range of matching thresholds, and instead compare the performance in terms of Area Under the Curve (AUC).

The PCK metric is also used to evaluate the performance of 2D hand pose estimation. Simon et al. (2017) use PCKh to calculate the threshold values. This, however, requires annotation of the head bounding box for each test image. To this end, Zimmermann and Brox (2017) use PCK with a fixed pixel value as threshold and do not perform any scale normalization.

3.3.1.3. Object Keypoint Similarity (OKS)

The PCK or PCKh metrics have the drawback that they treat all body keypoints in the same manner, *i.e.*, the threshold values for wrist and shoulders are the same. However, in general, a wrist covers a very small region of the body and should be localized precisely in contrast to a shoulder which covers a much larger region. Therefore, the threshold values should be adapted for each keypoint type, *i.e.*, a smaller threshold value for small keypoints and larger threshold values for the keypoints that encompass large regions. This is particularly important if the facial landmarks also have to be evaluated. To this end, Ronchi and Perona (2017) proposed the Keypoint Similarity (KS) metric which uses an un-normalized Gaussian distribution centered at the ground-truth location $\mathbf{x}_j$ of keypoint j, and measures the similarity by evaluating the distribution at the predicted location $\hat{\mathbf{x}}_j$ of the corresponding keypoint. Formally, it can be written as follows:

$$KS(\hat{\mathbf{x}}_j, \mathbf{x}_j) = e^{-\frac{||\hat{\mathbf{x}}_j - \mathbf{x}_j||_2^2}{2s^2\sigma_j^2}}, \tag{3.11}$$

where the Gaussian's standard deviation σ_j is specific to the keypoint type and is scaled by the ground-truth scale s. If the predicted location exactly matches the ground-truth, the value of the KS will be 1 and decreases toward zero as the distance increases. The OKS metric between the ground-truth pose $\mathbf{p}$ and the predicted pose $\hat{\mathbf{p}}$ is then computed as the average of the KS of all visible body joints as follows:

$$OKS(\hat{\mathbf{p}}, \mathbf{p}) = \frac{\sum_j KS(\hat{\mathbf{x}}_j, \mathbf{x}_j)\delta(v_j > 0)}{\sum_j \delta(v_j > 0)}, \tag{3.12}$$

where v_j indicates keypoints that are visible in the ground-truth annotation.

3.3.2. Multi-Person 2D Pose Estimation

PCK or OKS metrics are only suitable for single-person pose estimation with a known number of body joints, and do not penalize the false positives that are not part of the ground-truth. Therefore, for multi-person pose estimation the performance is evaluated in terms of the mean Average Precision (Pishchulin et al., 2016) as described next.

3.3.2.1. mean Average Precision (mAP)

The mAP is the mean of the Average Precision (AP) of all body joint types, where the AP for each joint type corresponds to the area under the smoothed recall-precision curve. Given a set of ground-truth and predicted poses, it starts by greedily assigning each predicted body pose to one of the GT poses. A pose can be assigned to only one GT. The body joints belonging to all unassigned poses are considered as False Positives (FP), while the joints from the remaining unassigned GT poses are regarded as False Negatives (FN). The joints from each GT-prediction pair are classified as True Positives (TP), FP, or FN using the PCK or OKS metric. The TP, FP, and FN are then used to generate the recall-precision curve and to compute the AP for each joint. In general, the threshold value for PCKh is taken as 50% of the head-segment length. Whereas for OKS, a threshold value of 0.5 is often used.

3.3.3. 3D Pose Estimation

3.3.3.1. Mean Per Joint Position Error (MPJPE)

The MPJPE is the most commonly used evaluation metric to evaluate 3D human body pose estimation. It measures the average Euclidean distance between the corresponding joints in the ground-truth 3D pose $\mathbf{P}$ and the predicted pose $\hat{\mathbf{P}}$, and is measured in millimeters (mm).

$$MPJPE = \frac{1}{J}\sum_{j=1}^{J} ||\mathbf{P}_j - \hat{\mathbf{P}}_j||_2 \tag{3.13}$$

MPJPE in the most strict form expects that the 3D poses are estimated in the camera coordinate system with absolute depth values. However, estimating the absolute depth values from an RGB image is an ill-posed problem, and many approaches instead predict the root-relative 3D pose. Therefore, it is a common practice in the literature to use relative-MPJPE which aligns the position of the root joints of the GT and predicted pose before computing MPJPE. Some approaches may also encounter difficulties in accurately predicting the limb-lengths or the scale of the person. Therefore, in some evaluation protocols, the ground-truth and the predicted poses are aligned using Procrustes analysis before computing MPJPE. The MPJPE is also referred to as End-Point-Error (EPE) or mean 3D joint error in the literature.

3.3.3.2. Mean Per Joint Localization Error (MPJLE)

MPJLE is a variant of PCK for 3D human pose estimation. It considers a 3D keypoint as correctly localized if it lies within a certain distance from the GT joint location. MPJLE is then computed as the percentage of correctly localized joints. Similar to MPJPE, root-normalization or pose alignment using Procrustes Analysis is often used before computing MPJLE.

3.3.4. Multi-Target Tracking

In this section we briefly explain the evaluation metrics used for multi-target tracking in general. In Chapter 5, we will propose an evaluation protocol for articulated multi-person pose tracking which will be built on these metrics.

3.3.4.1. CLEAR MOT

The CLEAR MOT evaluation protocol, proposed by Bernardin and Stiefelhagen (2008), is the widely used protocol in the tracking community. It encompasses two quantities, Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP). MOTA measures the tracking accuracy of the tracker, while MOTP assess its ability to localize the target in the image accurately.

The tracker is expected to output the locations of the targets over time with a unique track-id. To check whether a target is tracked correctly, the CLEAR MOT protocol first assigns each predicted track to one of the ground-truth tracks. This is done greedily by using an application specific metric, *e.g.*, for object tracking IoU is usually used, while for pose tracking KS or a ratio of the head-segment length may be used. However, the assignments are not done independently for each frame but are also propagated from the previous frames. In particular, a ground-truth track is assigned to the closest track, if and only if, it has not been assigned to another track in the previous frame. Otherwise, the correspondences from the previous frames are propagated.

Given the assignments, MOTA is computed based on a number of error ratios including False Positives (FP), False Negative (FN), and Identity Switches (IS). The FP correspond to erroneous tracking results that do not match to any ground-truth track. The FN are the tracks that are annotated in the ground-truth but are not tracked by the tracker. An IS is counted when a ground-truth target track is assigned to another track as compared to the previous frame. Let FP(f), FN(f) an IS(f) denote the number of false positives, false negatives and identity-switches at frame f, then the MOTA score is defined as

$$\text{MOTA} = 1 - \frac{\sum_f^F (\text{FP}(f) + \text{FN}(f) + \text{IS}(f))}{\sum_f^F N_{gt}(f)}, \tag{3.14}$$

where $N_{gt}(f)$ is the total number of GT targets to be tracked at frame f. It is important to note that the value of MOTA can become negative when the number of errors is larger than the number of ground-truths. An illustration of the errors used for MOTA can be seen in Fig 3.6.

MOTP measures how well a tracker has localized a target and is computed as follows:

$$\text{MOTP} = \frac{\sum_f^F \sum_i^{m_f} S(\mathbf{x}_i^f, \hat{\mathbf{x}}_i^f)}{\sum_f^F m_f}, \tag{3.15}$$

where $\mathbf{x}_i^f$ and $\hat{\mathbf{x}}_i^f$ are the locations of a GT track and its associated prediction at frame f, respectively, and m_f is the total number of matches at frame f. The function $S(.)$ is the similarity metric used to build the correspondences between the GT and predicted tracks.

3.3.4.2. Trajectory-based measures

The trajectory-based metrics, proposed by Li et al. (2009), count the number of mostly tracked (MT), mostly lost (ML), and partially tracked (PT) tracks. A track is considered mostly tracked if the tracker

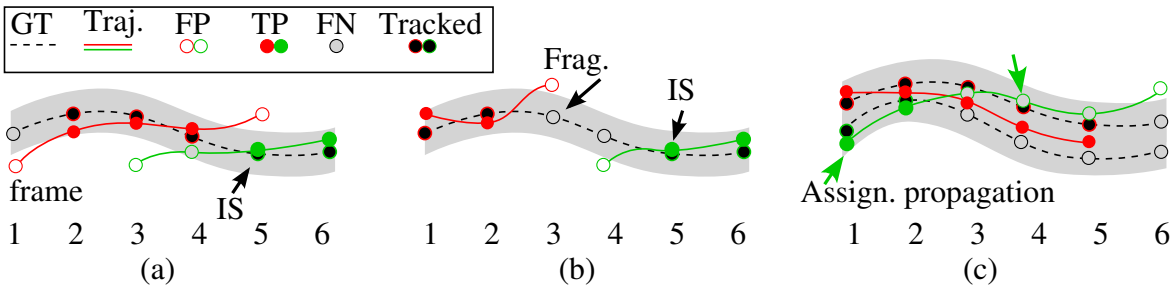


Figure 3.6: Illustrations of the tracker-to-target assignments and example error cases. (a) An identity switch (IS) is counted at frame 5 when the assignment switches from the previously assigned red track to the green track. (b) A track fragmentation occurs in frame 3 since the target is tracked from frame 1 to frame 2, then breaks at frame 3, and is tracked again from frame 5. The new track (green) also result in an IS at this point. (c) demonstrates an example of assignment propagation. Both green and red tracks are greedily assigned to one of the GTs at frame 1. At frame 3, although the green track is more closer to the first GT trajectory, the optimal single-frame assignment from frame 1 is propagated through the sequence, resulting in 4 false positives (FP) and 5 missed target (FN). Note that no fragmentations occur in frames 3 and 6 because the tracking of those targets never resumes later. The figure is adopted from (Milan et al., 2016).

has found it for at least 80% of its length, and mostly lost if it is not correctly tracked for 80% of its length. All remaining tracks are considered as partially tracked. In addition, the count of track fragmentations (FM) is also reported. A fragmentation is counted everytime a GT track changes its status from “tracked” to “not tracked”. In contrast to CLEAR MOT metrics, all trajectory-based metrics are measured on entire trajectories instead of frame-by-frame basis.

CHAPTER 4

Multi-Person 2D Pose Estimation from Images

In this chapter, we propose a method that estimates the poses of multiple persons in an image in which a person can be occluded by another person or might be truncated. To this end, we consider multi-person pose estimation as a joint-to-person association problem. We construct a fully connected graph from a set of detected joint candidates in an image and resolve the joint-to-person association and outlier detection using integer linear programming. Since solving joint-to-person association jointly for all persons in an image is an NP-hard problem and even approximations are expensive, we solve the problem locally for each person. On the challenging MPII Human Pose Dataset for multiple persons, our approach achieves the accuracy of a state-of-the-art method, but it is 6,000 times faster.

Contents

4.1. INTRODUCTION

Single person pose estimation has made a remarkable progress over the past few years. This is mainly due to the availability of deep learning based methods for detecting joints (Carreira et al., 2016; Pishchulin et al., 2016; Wei et al., 2016; Tompson et al., 2015; Insafutdinov et al., 2016). While earlier approaches in this direction (Chen and Yuille, 2014; Tompson et al., 2014b, 2015) combine the body part detectors with tree structured graphical models, more recent methods (Carreira et al., 2016; Pishchulin et al., 2016; Wei et al., 2016; Newell et al., 2016; Bulat and Tzimiropoulos, 2016; Rafi et al., 2016) demonstrate that spatial relations between joints can be directly learned by a neural

Figure 4.1: Example image from the multi-person subset of the MPII Pose Dataset (Andriluka et al., 2014).

network without the need of an additional graphical model. These approaches, however, assume that only a single person is visible in the image and the location of the person is known a-priori. Moreover, the number of parts are defined by the network, *e.g.*, full body or upper body, and cannot be changed. For realistic scenarios such assumptions are too strong and the methods cannot be applied to images that contain a number of overlapping and truncated persons. An example of such a scenario is shown in Figure 4.1.

In comparison to single person human pose estimation benchmarks, multi-person pose estimation introduces new challenges. The number of persons in an image is unknown and needs to be correctly estimated, the persons occlude each other and might be truncated, and the joints need to be associated to the correct person. The simplest approach to tackle this problem is to first use a person detector and then estimate the pose for each detection independently (Pishchulin et al., 2012; Gkioxari et al., 2014; Chen and Yuille, 2015). This, however, does not resolve the joint association problem of two persons next to each other or truncations. Other approaches estimate the pose of all detected persons jointly (Eichner and Ferrari, 2010; Ladicky et al., 2013). In (Pishchulin et al., 2016) a person detector is not required. Instead body part proposals are generated and connected in a large graph. The approach then solves the labeling problem, the joint-to-person association problem and non-maximum suppression jointly. While the model proposed in (Pishchulin et al., 2016) can be solved by integer linear programming and achieves state-of-the-art results on a very small subset of the MPII Human Pose Dataset, the complexity makes it infeasible for a practical application. As reported in (Insafutdinov et al., 2016), the processing of a single image takes about 72 hours.

In this chapter, we address the joint-to-person association problem using a densely connected graphical model as in (Pishchulin et al., 2016), but propose to solve it only locally. To this end, we first use a person detector and crop image regions as illustrated in Figure 4.1. Each of the regions contains sufficient context, but only the joints of persons that are very close. We then solve the joint-to-person association for the person in the center of each region by integer linear programming (ILP). The labeling of the joints and non-maxima suppression are directly performed by a convolutional neural network. We evaluate our approach on the MPII Human Pose Dataset for multiple persons where we slightly improve the accuracy of (Pishchulin et al., 2016) while reducing the runtime by a factor between 6,000 and 19,000.

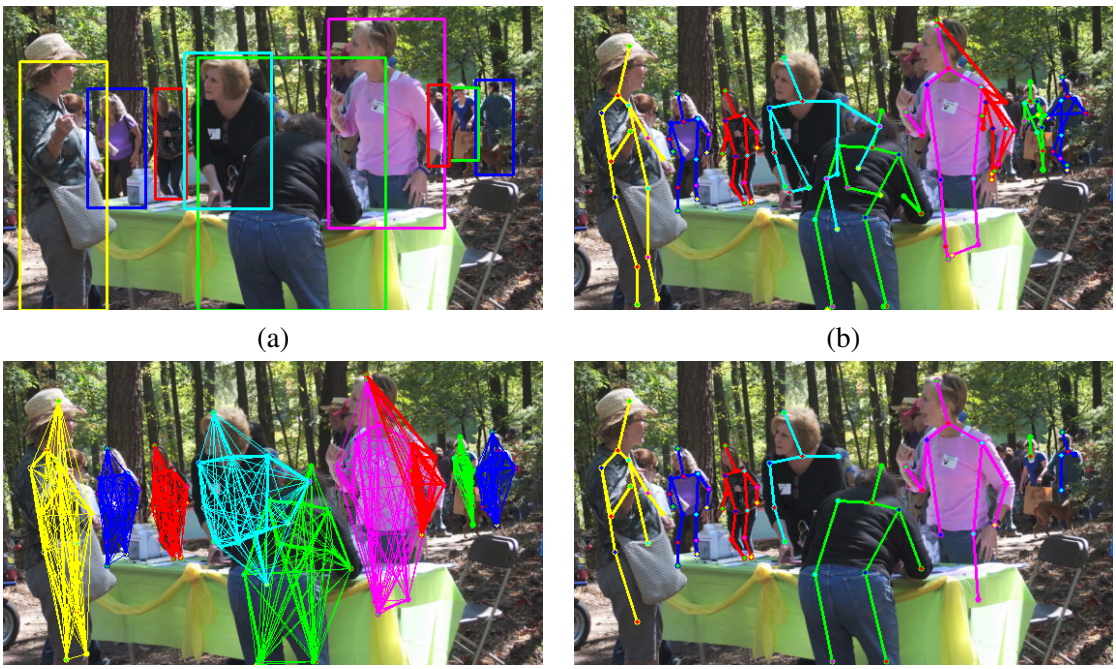


Figure 4.2: Overview of the proposed method. We detect persons in an image using a person detector (a). A set of joint candidates is generated for each detected person (b). The candidates build a fully connected graph (c) and the final pose estimates are obtained by integer linear programming (d). (best viewed in color)

4.2. OVERVIEW

Our method solves the problem of joint-to-person association locally for each person in the image. To this end, we first detect the persons using a person (Ren et al., 2015). For each detected person, we generate a set of joint candidates using a single person pose estimation model (Section 4.3). The candidates are prone to errors since the single person models do not take into account occlusion or truncation. In order to associate each joint to the correct person and also to remove the erroneous candidates, we perform inference locally on a fully connected graph for each person using integer linear programming (Section 4.4). Figure 4.2 shows an overview of the proposed approach.

4.3. CONVOLUTIONAL POSE MACHINES

Given a person in an image $\mathbf{I}$, we define its 2D pose as a set $\mathbf{p} = \{\mathbf{x}_j\}_{j \in \mathcal{J}}$, where the vector $\mathbf{x}_j \in \mathbf{p}$ represents the 2D location (x, y) of the body joint of type $j \in \mathcal{J} = \{1, \ldots, J{=}14\}$ in the image. The convolutional pose machines consist of a multi-staged CNN architecture with $k \in \{1 \ldots K\}$ stages, where each stage is a multi-label classifier $\Phi_k(\mathbf{x})$ that is trained to provide confidence maps $H_k^j \in \mathbb{R}^{w \times h}$ for each joint $j \in \mathcal{J}$ and the background, where w and h are the width and the height of the image, respectively.

The first stage of the architecture uses only the local image evidence and provides the confidence

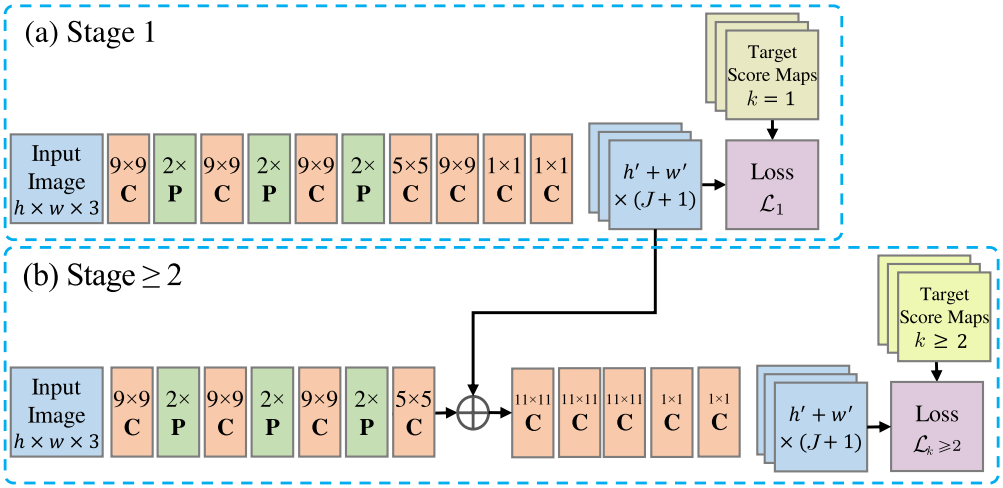


Figure 4.3: CPM architecture proposed in (Wei et al., 2016). The first stage (a) utilizes only the local image evidence whereas all subsequent stages (b) also utilize the output of preceding stages to exploit the spatial context between joints. The receptive field of stages $k \geqslant 2$ is increased by having multiple convolutional layers at the 8 times down-sampled score maps. All stages are locally supervised and a separate loss is computed for each stage. We provide multi-person target score maps to stage 1, and single-person score maps to all subsequent stages.

scores

$$\Phi_{k=1}(\mathbf{x}|\mathbf{I}) \rightarrow \{H_1^j(\mathbf{x}_j = \mathbf{x})\}_{j=1\ldots J+1}. \tag{4.1}$$

Whereas, in addition to the local image evidence, all subsequent stages also utilize the contextual information from the preceding stages to produce confidence score maps

$$\Phi_{k>1}(\mathbf{x}|\mathbf{I}, \xi(\mathbf{x}, \mathbf{H_{k-1}})) \rightarrow \{H_k^j(\mathbf{x}_j = \mathbf{x})\}_{j=1\ldots J+1}, \tag{4.2}$$

where $\mathbf{H}_k \in \mathbb{R}^{w \times h \times (J+1)}$ corresponds to the score maps of all body joints and the background at k^{th} stage, and $\xi(\mathbf{x}, \mathbf{H}_{k-1})$ indicates the mapping from the scores $\mathbf{H}_{k-1}$ to the context features for location $\mathbf{x}$. The receptive field of the subsequent stages is increased to the extent that the context of the complete person is available. This allows to model complex long-range spatial relationships between joints, and to leverage the context around the person. The CPM architecture is completely differentiable and allows end-to-end training of all stages. Due to the multi-stage nature of CPM, the overall CNN architecture consists of many layers and is therefore prone to the problem of vanishing gradients (Bengio et al., 1994; Glorot and Bengio, 2010; Wei et al., 2016). In order to solve this problem, Wei et al. (2016) proposed to use intermediate supervision by adding a loss function at each stage k. The CNN architecture used for each stage can be seen in Figure 4.3. In this work we exploit the intermediate supervision of the stages during training for multi-person human pose estimation as we will discuss in the next section.

4.3.1. Training for Multi-Person Pose Estimation

Each stage of the CPM is trained to produce confidence score maps for all body joints, and the loss function at the end of every stage computes the l_2 distance between the predicted confidence scores

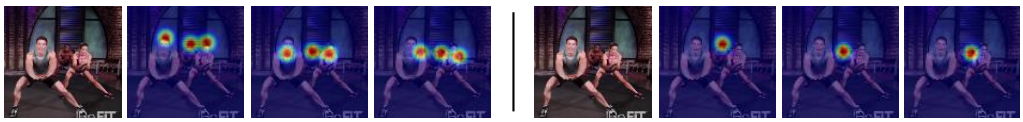


Figure 4.4: Example of target score maps for the head, neck and left shoulder. The target score maps for the first stage include the joints of all persons (left). The target score maps for all subsequent stages only include the joints of the primary person.

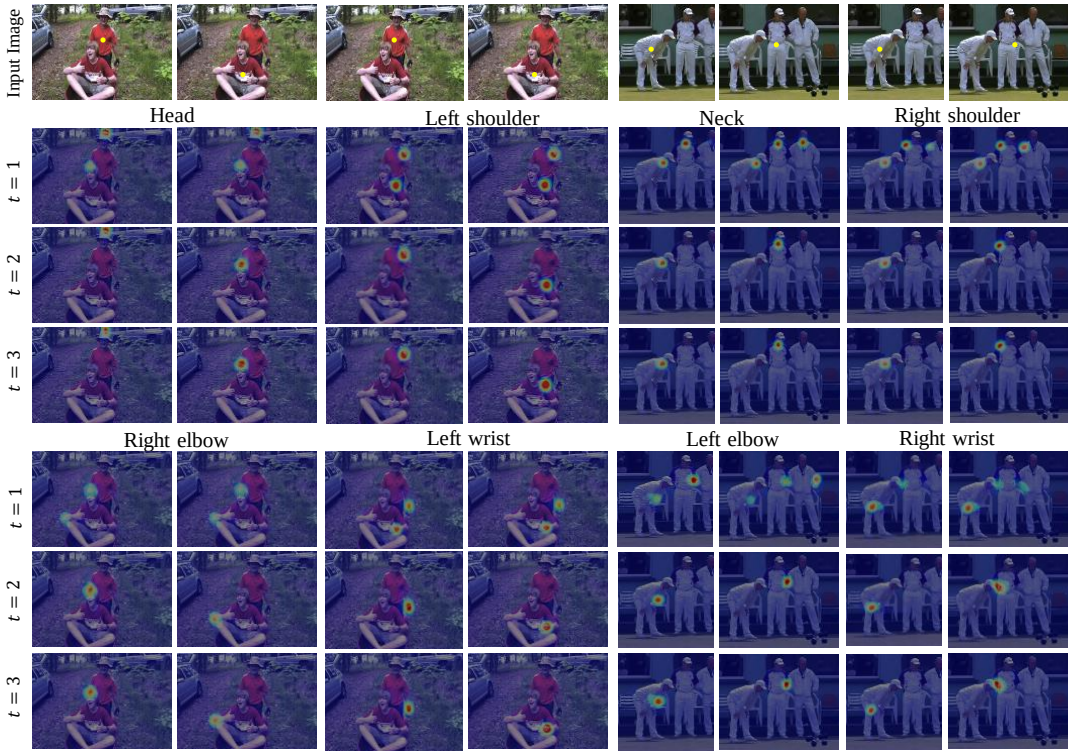


Figure 4.5: Examples of score maps provided by different stages of the CPM. The first stage of CPM uses only local image evidence and therefore provides high confidence scores for the joints of all persons in the image. Whereas all subsequent stages are trained to provide high confidence scores only for the joints of the primary person while suppressing the joints of other persons. The primary person is highlighted by a yellow dot in the first row. (best viewed in color)

and the target score maps. The target score maps are modeled as Gaussian distributions centered at the ground-truth locations of the joints. For multi-person pose estimation, the aim of the training is to focus only on the body joints of the detected person, while suppressing joints of all other overlapping persons. We do this by creating two types of target score maps. For the first stage, we model the target score maps by a sum of Gaussian distributions for the body joints of all persons appearing in the bounding box enclosing the primary person that appears roughly in the center of the bounding box. For the subsequent stages, we model only the joints of the primary person. An example of target score maps for different stages can be seen in Figure 4.4.

Figure 4.5 shows some examples how the inferred score maps evolve as the number of stages increases. In (Wei et al., 2016), the pose of the person is obtained by taking the maximum of the

inferred score maps, *i.e.*, $\mathbf{x}_j = \arg\max_{\mathbf{x}} H_K^j(\mathbf{x})$.

This, however, assumes that all joints are visible in the image and results in erroneous estimates for invisible joints and can wrongly associate joints of other nearby persons to the primary person. Instead of taking the maximum, we sample N candidates from each inferred score map H_K^j and resolve the joint-to-person association and outlier removal by integer linear programming.

4.4. JOINT-TO-PERSON ASSOCIATION

We solve the joint-to-person association using a densely connected graphical model as in (Pishchulin et al., 2016). The model proposed in (Pishchulin et al., 2016), however, aims to resolve joint-to-person associations together with proposal labeling globally for all persons, which makes it very expensive to solve. In contrast, we propose to solve this problem locally for each person. We first briefly revise the DeepCut method (Pishchulin et al., 2016) in Section 4.4.1, and then describe the proposed local joint-to-person association model in Section 4.4.2.

4.4.1. DeepCut

DeepCut aims to solve the problem of multi-person human pose estimation by jointly modeling the poses of all persons appearing in an image. Given an image, it starts by generating a set D of joint proposals, where $\mathbf{x}_d \in \mathbb{R}^2$ denotes the 2D location of the d^{th} proposal. The proposals are then used to formulate a graph optimization problem that aims to select a subset of proposals while suppressing the incompatible proposals, label each selected proposal with a joint type $j \in \mathcal{J}$, and associate them to unique individuals.

The problem can be solved by integer linear programming (ILP), optimizing over the binary variables $v \in \{0,1\}^{|D|\times|\mathcal{J}|}$, $s \in \{0,1\}^{\binom{|D|}{2}}$, and $t \in \{0,1\}^{\binom{|D|}{2}\times|\mathcal{J}|^2}$. For every proposal d, a set of variables $\{v_{d_j}\}_{j\in\mathcal{J}}$ is defined where $v_{d_j} = 1$ indicates that the proposal d is of body joint type j. For every pair of proposals (d, d'), the variable $s_{(d,d')}$ indicates that the proposals d and d' belong to the same person. The variable $t_{(d_j,d'_{j'})} = 1$ indicates that the proposal d is of joint type j, the proposal d' is of joint type j', and both proposals belong to the same person *i.e.*, $s_{(d,d')} = 1$. The variable $t_{(d_j,d'_{j'})}$ is constrained such that $t_{(d_j,d'_{j'})} = v_{d_j} v_{d'_{j'}} s_{(d,d')}$. The solution of the ILP problem is obtained by optimizing the following objective function:

$$\arg\min_{v,s,t} \langle v, \phi\rangle + \langle t, \psi\rangle \tag{4.3}$$

where

$$\langle \phi, v\rangle = \sum_{d\in D}\sum_{j\in\mathcal{J}} v_{dj}\phi_{dj} \tag{4.4}$$

$$\langle \psi, t\rangle = \sum_{(d,d')\in D}\sum_{(j,j')\in\mathcal{J}} t_{(d_j,d'_{j'})}\psi_{(d_j,d'_{j'})}. \tag{4.5}$$

This means that we search for the solution that minimizes the cost of the nodes and edges. The cost to set the variable $v_{d_j} = 1$ is defined by the unary term:

$$\phi_{dj} = \log\frac{1-p_{d_j}}{p_{d_j}} \tag{4.6}$$

where $p_{d_j} \in (0, 1)$ corresponds the probablity that the proposal d is of joint type j. Note that ϕ_{dj} is negative when $p_{dj} > 0.5$ and therefore the detection with high confidence are preferred since they reduce the cost function (4.3). The cost for binary potentials is defined similarly by:

$$\psi_{(d_j,d'_{j'})} = \log \frac{1 - p_{(d_j,d'_{j'})}}{p_{(d_j,d'_{j'})}} \tag{4.7}$$

where $p_{(d_j,d'_{j'})}$ corresponds to the conditional probability that a pair of proposals (d, d') belongs to the same person, given that d and d' are of joint type j and j', respectively. In order to ensure the solution of the objective (4.3) results in plausible body poses and joint type assignments, the problem is optimized subject to the additional constraints:

$$v_{d_j} + v_{d_{j'}} \leqslant 1 \quad \forall d \in D \quad \forall j \in \mathcal{J} \tag{4.8}$$

$$s_{(d,d')} \leqslant \sum_{j \in \mathcal{J}} v_{d_j}, \quad s_{(d,d')} \leqslant \sum_{j \in \mathcal{J}} v_{d'_j} \quad \forall (d, d') \in D \tag{4.9}$$

$$s_{(d,d')} + s_{(d',d'')} - 1 \leqslant s_{(d,d'')} \quad \forall (d, d', d'') \in D \tag{4.10}$$

$$\begin{aligned} v_{d_j} + v_{d'_{j'}} + s_{(d,d')} - 2 &\leqslant t_{(d_j,d'_{j'})} \\ t_{(d_j,d'_{j'})} &\leqslant min(v_{d_j}, v_{d'_{j'}}, s_{(d,d')}) \\ \forall (d, d') \in D \quad \forall (j, j') &\in \mathcal{J} \end{aligned} \tag{4.11}$$

and optinally,

$$v_{d_j} + v_{d'_{j'}} - 1 \leqslant s_{(d,d')} \quad \forall (d, d') \in D \; \forall (j, j') \in \mathcal{J} \tag{4.12}$$

The constraints (4.8)-(4.11) enforce that optimizing (4.3) results in valid body pose configurations for one or more persons. The constraints (4.8) ensure that a proposal d can be labeled with only one joint type, while the constraints (4.9) guarantee that any pair of proposals (d, d') can belong to the same person only if both are not suppressed, *i.e.*, $v_{d_j} = 1$ and $v_{d'_{j'}} = 1$. The constraints (4.10) are transitivity constraints and enforce for any triplet of proposals $(d, d', d'') \in D$ that if d and d' belong to the same person, and d' and d'' also belong to the same person, then the proposals d and d'' must also belong to the same person. The constraints (4.11) enforce that for any $(d, d') \in D$ and $(j, j') \in \mathcal{J}$, $t_{(d_j,d'_{j'})} = v_{d_j} v_{d'_{j'}} s_{(d,d')}$. The constraints (4.12) are only applicable for single-person human pose estimation, as they enforce that two proposals (d, d') that are not suppressed must be grouped together. In (Pishchulin et al., 2016) this ILP formulation is referred as *Subset Partitioning and Labelling Problem*, as it partitions the initial pool of proposal candidates to unique individuals, labels each proposal with a joint type j, and inherently suppresses the incompatible candidates.

4.4.2. Local Joint-to-Person Association

In contrast to (Pishchulin et al., 2016), we solve the joint-to-person association problem locally for each person. We also do not label generic proposals as part of the ILP formulation since we use a neural network to obtain detections for each joint as described in Section 4.3. We therefore start with a set of joint detections D_J, where every detection d at location $\mathbf{x}_d \in \mathbb{R}^2$ has a known joint type $j \in \mathcal{J}$. Our model requires only two types of binary random variables $v \in \{0,1\}^{|D_J|}$ and $s \in \{0,1\}^{\binom{D_J}{2}}$. Here, $v_d = 1$ indicates that the detection d of part type j is not suppressed, and $s_{(d,d')} = 1$ indicates that the detection d of type j, and the detection d' of type j' belong to the same person. We drop the j from v_{d_j} here for notational clarity. The objective function for local joint-to-person association takes the form:

$$\arg\min_{v,s} \langle v, \phi \rangle + \langle s, \psi \rangle \tag{4.13}$$

subject to

$$s_{(d,d')} \leqslant v_d, \quad s_{(d,d')} \leqslant v_{d'} \quad \forall (d,d') \in D_J \tag{4.14}$$

$$s_{(d,d')} + s_{(d',d'')} - 1 \leqslant s_{(d,d'')} \quad \forall (d,d',d'') \in D_J \tag{4.15}$$

$$v_d + v_{d'} - 1 \leqslant s_{(d,d')} \quad \forall (d,d') \in D_J \tag{4.16}$$

where

$$\langle \phi, v \rangle = \sum_{d \in D_J} v_d \phi_d \tag{4.17}$$

$$\langle \psi, s \rangle = \sum_{(d,d') \in D_J} s_{(d,d')} \psi_{(d,d')}. \tag{4.18}$$

$$\phi_d = \log \frac{1 - p_d}{p_d} \tag{4.19}$$

$$\psi_{(d,d')} = \log \frac{1 - p_{(d,d')}}{p_{(d,d')}} \tag{4.20}$$

The constraints (4.14) enforce that detection d and d' are connected ($s_{(d,d')} = 1$) only if both are not suppressed, *i.e.*, $v_d = 1$ and $v_{d'} = 1$. The constraints (4.15) are transitivity constraints as before and the constraints (4.16) guarantee that all detections that are not suppressed belong to the primary person. We can see from (4.3)-(4.12) and (4.13)-(4.16), that the number of variables are reduced from $(|D| \times |\mathcal{J}| + \binom{|D|}{2} + \binom{|D|}{2} \times |\mathcal{J}|^2)$ to $(|D_J| + \binom{|D_J|}{2})$. Similary, the number of constraints is also drastically reduced.

In (4.19), $p_d \in (0,1)$ is the confidence of the joint detection d as probability. We obtain this directly from the score maps inferred by the CPM as $p_d = \mathrm{f}_\tau(H_K^j(\mathbf{x}_d))$, where

$$\mathrm{f}_\tau(H) = \begin{cases} H & \text{if} \quad H \geqslant \tau \\ 0 & \text{otherwise,} \end{cases} \tag{4.21}$$

and τ is a threshold that suppresses detections with a low confidence score.

In (4.20), $p_{(d,d')} \in (0,1)$ corresponds to the conditional probability that the detection d of joint type j and the detection d' of joint type j' belong to the same person. For $j = j'$, it is the probability that both detections d and d' belong to the same body joint. For $j \neq j'$, it measures the compatibility between two detection candidates of different joint types. Similar to (Pishchulin et al., 2016), we obtain these probabilities by learning discriminative models based on appearance and spatial features of the detection candidates. For $j = j'$, we define a feature vector

$$\mathbf{g}_{(d,d')} = \{\triangle\mathbf{x}, \exp(\triangle\mathbf{x}), (\triangle\mathbf{x})^2\}, \tag{4.22}$$

where $\triangle\mathbf{x} = (\triangle x, \triangle y)$ is the 2D offset between the locations $\mathbf{x}_d$ and $\mathbf{x}_{d'}$. For $j \neq j'$, we define a separate feature vector based on the spatial locations as well as the appearance features obtained from the joint detectors as

$$\mathbf{g}_{(d,d')} = \{\triangle\mathbf{x}, \|\triangle\mathbf{x}\|, \arctan\left(\frac{\triangle y}{\triangle x}\right), \mathbf{H}_K(\mathbf{x}_d), \mathbf{H}_K(\mathbf{x}_{d'})\}, \tag{4.23}$$

where $\mathbf{H}_K(\mathbf{x})$ is a vector containing the confidences of all joints and the background at location $\mathbf{x}$. For both cases, we gather positive and negative samples from the annotated poses in the training data and train an SVM with RBF kernel using LibSVM (Chang and Lin, 2011) for each pair $(j, j') \in \mathcal{J}$. In order to obtain the probabilities $p_{(d,d')} \in (0,1)$ we use Platt scaling (Platt, 1999) to normalize the output of the SVMs to probabilities. After optimizing (4.13), the pose of the primary person is given by the detections with $v_d = 1$.

4.5. EXPERIMENTS

We evaluate the proposed approach on the Multi-Person subset of the MPII Human Pose Dataset (Andriluka et al., 2014) and follow the evaluation protocol proposed in (Pishchulin et al., 2016). The dataset consists of 3844 training and 1758 testing images with multiple persons. The persons appear in highly articulated poses with a large amount of occlusions and truncations. Since the original test data of the dataset is withheld, we perform all intermediate experiments on a validation set of 1200 images. The validation set is sampled according to the split proposed in (Tompson et al., 2015) for the single person setup, *i.e.*, we chose all multi-person images that are part of the validation test set proposed in (Tompson et al., 2015) and use all other images for training. In addition we compare the proposed method with our main competitor approach (Pishchulin et al., 2016) on their selected subset of 288 images We also provide a comparison with the current state-of-the-art approaches on the complete test set. The accuracy is measured by average precision (AP) for each joint using the scripts provided by (Pishchulin et al., 2016).

4.5.1. Implementation Details

In order to localize the persons, we use the person detector proposed in (Ren et al., 2015). The detector is trained on the Pascal VOC dataset (Everingham et al., 2015). For the quantitative evaluation, we discard detected persons with a bounding box area less equal to 80×80 pixels since small persons are not annotated in the MPII Human Pose Dataset. For the qualitative results shown in Figure 4.7, we do not discard the small detections. For the CPM (Wei et al., 2016), we use the publicly available source code and train it on the Multi-Person subset of the MPII Human Pose Dataset as described

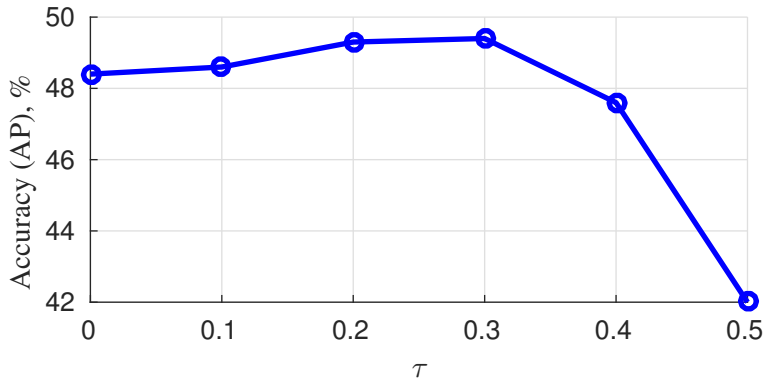


Figure 4.6: Impact of the parameter τ in (4.21) on the pose estimation accuracy.

Setting	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	time (s)
CPM only	56.5	55.2	47.6	39.1	48.1	39.6	30.3	45.2	2
CPM+L-JPA (N=1)	59.9	57.5	50.5	41.8	49.8	45.5	39.5	49.2	3
CPM+L-JPA (N=3)	59.9	57.4	50.7	42.1	50.0	45.5	39.5	49.3	8
CPM+L-JPA (N=5)	60.0	57.5	50.7	42.1	50.1	45.5	39.4	49.3	10
CPM+L-JPA (N=5)+GT Torso	92.9	91.3	78.4	61.8	81.0	71.4	61.8	76.9	10

Table 4.1: Pose estimation results (AP) on the validation test set (1200 images) of the MPII Multi-Person Pose Dataset.

in Section 4.3. As in (Wei et al., 2016), we add images from the Leeds Sports Dataset (Johnson and Everingham, 2010b) during training, and use a 6 stage ($K = 6$) CPM architecture. For solving (4.13), we use the Gurobi Optimizer.

4.5.2. Results

We first evaluate the impact of the parameter τ in $f_\tau(s)$ (4.21) on the pose estimation accuracy measured as mean AP on the validation set containing 1200 images. Figure 4.6 shows that the function f_τ improves the accuracy when τ is increased until $\tau = 0.3$. For $\tau > 0.4$, the accuracy drops since a high value discards correct detections. For the following experiments, we use $\tau = 0.2$.

Table 4.1 reports the pose estimation results under different settings of the proposed approach on the validation set. We also report the median run-time required by each setting[1]. Using only the CPM to estimate the pose of each detected person achieves 45.2% mAP and takes only 2 seconds per image. Using the proposed Local Joint-to-Person Association (L-JPA) model with 1 detection candidate per joint ($N = 1$) to suppress the incompatible detections improves the performance from 45.2% to 49.2% with a very slight increase in run-time. Increasing the number of candidates per joint increases the accuracy only slightly. For the following experiments, we use $N = 5$. When we compare the numbers with Figure 4.6, we observe that CPM+L-JPA outperforms CPM for any $0 \leqslant \tau \leqslant 0.4$.

The accuracy also depends on the used person detector. We use an off-the-shelf person detector without any fine-tuning on the MPII dataset. In order to evaluate the impact of the person detector accuracy, we also estimate poses when the person detections are given by the ground-truth torso (GT

[1]Measured on a 2GHz Intel(R) Xeon(R) CPU with a single core and NVidia Geforce GTX Titan-X GPU.

Setting	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	time (s)
Ours	70.0	65.2	56.4	46.1	52.7	47.9	44.5	54.7	10
DeepCut (Pishchulin et al., 2016)	73.1	71.7	58.0	39.9	56.1	43.5	31.9	53.5	57995
Ours GT ROI	87.7	81.6	68.9	56.1	66.4	59.4	54.0	67.7	10
DeepCut GT ROI (Pishchulin et al., 2016)	78.1	74.1	62.2	52.0	56.9	48.7	46.1	60.2	57995
Chen et al., GT ROI (Chen and Yuille, 2014)	65.0	34.2	22.0	15.7	19.2	15.8	14.2	27.1	-
DeeperCut (Insafutdinov et al., 2016)	87.9	84.0	71.9	63.9	68.8	63.8	58.1	71.2	230
Newell et al. (2017)	91.5	87.2	75.9	65.4	72.2	67.0	62.1	74.5	1.47
Varadarajan et al. (2017)	92.9	88.8	77.7	67.8	74.6	67.0	63.8	76.1	-
Cao et al. (2017)	92.9	91.3	82.3	72.6	76.0	70.9	66.8	79.0	1.24
Fang et al. (2017)	89.3	88.1	80.7	75.5	73.7	76.7	70.0	79.1	1.5
Insafutdinov et al. (2017)	92.2	91.3	80.8	71.4	79.1	72.6	67.8	79.3	-

Table 4.2: Comparison of pose estimation results (AP) with state-of-the-art approaches on 288 images (Pishchulin et al., 2016).

Torso) locations of the persons provided with the dataset. This results in a significant improvement in accuracy from 49.3% to 76.9% mAP, showing that a better person detector would improve the results further.

Table 4.2 compares the proposed approach with other approaches on a selected subset of 288 test images used in (Pishchulin et al., 2016). Our approach outperforms our direct competitor method DeepCut (Pishchulin et al., 2016) (54.7% vs. 53.5%) while being significantly faster (10 seconds vs. 57995 seconds). If we use $N = 1$, our approach requires only 3 seconds per image with a minimal loss of accuracy as shown in Table 4.1, *i.e.*, our approach is more than 19,000 times faster than (Pishchulin et al., 2016). We also compare with (Pishchulin et al., 2016; Chen and Yuille, 2014) when using GT bounding boxes of the persons. The results for (Chen and Yuille, 2014) are taken from (Pishchulin et al., 2016). Our approach outperforms both methods by a large margin.

We also compare our results with the approaches that were published concurrently (Insafutdinov et al., 2016) or after (Varadarajan et al., 2017; Cao et al., 2017; Fang et al., 2017; Insafutdinov et al., 2017) our work. While the performance of recent methods increased rapidly due to stronger CNN models, and data augmentation techniques, our work provided first steps toward achieving this goal. For example, Insafutdinov et al. (2017) built on our simplified optimization problem in Section 4.4.2 and significantly reduced the runtime of their previous approaches (Pishchulin et al., 2016; Insafutdinov et al., 2016). Moreover,in contrast to the recent methods, we do not perform fine-tuning of the person detector on the MPII Multi-Person Pose Dataset and envision that doing this will lead to further improvements.

Finally in Table 4.3, we report our results on all test images of the MPII Multi-Person Pose Dataset. Our method achieves 43.1% mAP. While the approach (Pishchulin et al., 2016) cannot be evaluated on all test images due to the high computational complexity of the model. The concurrent approach (Insafutdinov et al., 2016) reports a higher accuracy than our model. However, if we compare the run-times in Table 4.2 and Table 4.3, we observe that the run-time of (Insafutdinov et al., 2016) doubles on the more challenging test set (485 seconds per image). Our approach on the other hand requires only 10 seconds in all evaluation settings and is around 50 times faster. If we use $N = 1$, our approach is 160 times faster than (Insafutdinov et al., 2016). Using the torso annotation (GT Torso) as person detections results again in a significant improvement of the accuracy (62.2% vs. 43.1% mAP). Some qualitative results can be seen in Figure 4.7.

Figure 4.7: Some qualitative results for the MPII Multi-Person Pose Dataset.

Setting	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	time (s)
Ours	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
DeeperCut (Insafutdinov et al., 2016)	78.4	72.5	60.2	51.0	‘ 57.2	52.0	45.4	59.5	485
Using GT ROIs									
Ours + GT Torso	85.6	79.4	62.9	48.9	62.6	51.9	43.9	62.2	10
State-of-the-art approaches published later.									
Levinkov et al. (2017)	89.8	85.2	71.8	59.6	71.1	63.0	53.5	70.6	-
Varadarajan et al. (2017)	92.1	85.9	72.9	61.7	72.0	64.6	56.6	72.2	-
Insafutdinov et al. (2017)	88.8	87.0	75.9	64.9	74.2	68.8	60.5	74.3	-
Cao et al. (2017)	91.2	87.6	77.7	66.8	75.4	68.9	61.7	75.6	1.24
Fang et al. (2017)	88.4	86.5	78.6	70.4	74.4	73.0	65.8	76.7	1.5
Newell et al. (2017)	92.1	89.3	78.9	69.8	76.2	71.6	64.7	77.5	1.47

Table 4.3: Pose estimation results (AP) on the withheld test set of the MPII Multi-Person Pose Dataset.

4.6. SUMMARY

In this chapter we have presented an approach for multi-person pose estimation under occlusions and truncations. Since the global modeling of poses for all persons is impractical, we demonstrated that the problem can be formulated by a set of independent local joint-to-person association problems. Compared to global modeling, these problems can be solved efficiently while still being effective for handling severe occlusions or truncations. Although the accuracy can be further improved by using a better person detector and stronger CNN models, the proposed method already achieves the accuracy of a global modeling based approach, while being 6,000 to 19,000 times faster.

CHAPTER 5

Joint Multi-Person Pose Estimation and Tracking in Videos

In Chapter 4 we presented an approach that can estimate the poses of multiple people in unconstrained images. The proposed method does not make any assumption regarding the appearance of the people, nor does it assume that the number of people is known. However, it cannot be applied to videos directly since, in this case, we also need to solve the problem of person association over time in addition to the pose estimation for each person.

Therefore, in this chapter, we introduce the challenging problem of joint multi-person pose estimation and tracking of an unknown number of persons in unconstrained videos. We propose a novel method that jointly models multi-person pose estimation and tracking in a single formulation. Since the problem has not been addressed quantitatively in the literature, we introduce a challenging "Multi-Person Pose-Track" dataset and also propose an unconstrained evaluation protocol that does not make any assumptions on the scale, size, location or the number of persons. Finally, we evaluate the proposed approach and several baseline methods on our new dataset.

Contents

5.1. INTRODUCTION

The field of human pose estimation in images has progressed remarkably over the past few years. The methods have advanced from pose estimation of single pre-localized persons (Pishchulin et al., 2016;

Carreira et al., 2016; Wei et al., 2016; Hu and Ramanan, 2016; Insafutdinov et al., 2016; Newell et al., 2016; Bulat and Tzimiropoulos, 2016; Rafi et al., 2016) to the more challenging and realistic case of multiple, potentially overlapping and truncated persons (Gkioxari et al., 2014; Chen and Yuille, 2015; Pishchulin et al., 2016; Insafutdinov et al., 2016; Iqbal and Gall, 2016). Many applications, such as mentioned before, however, aim to analyze human body motion over time. While there exists a notable number of works that track the pose of a single person in a video (Park and Ramanan, 2011; Cherian et al., 2014; Zhang and Mubarak, 2015; Ramakrishna et al., 2013; Zuffi et al., 2013; Jain et al., 2014b; Pfister et al., 2015; Charles et al., 2016; Gkioxari et al., 2016; Iqbal et al., 2017a), multi-person human pose estimation in unconstrained videos has not been addressed in the literature.

In this chapter, we address the problem of tracking the poses of multiple persons in an unconstrained setting. This means that we have to deal with large pose and scale variations, fast motions, and a varying number of persons and visible body parts due to occlusion or truncation. In contrast to previous works, we aim to solve the association of each person across the video and the pose estimation together. To this end, we build upon our approach from Chapter 4 and recent methods for multi-person pose estimation in images (Pishchulin et al., 2016; Insafutdinov et al., 2016) that also build a spatial graph based on joint proposals to estimate the pose of multiple persons in an image. In particular, we cast the problem as an optimization of a densely connected spatio-temporal graph connecting body joint candidates spatially as well as temporally. The optimization problem is formulated as a constrained Integer Linear Program (ILP) whose feasible solution partitions the graph into valid body pose trajectories for any unknown number of persons. In this way, we can handle occlusion, truncation, and temporal association within a single formulation.

Previous datasets used to benchmark pose estimation algorithms in-the-wild are summarized in Table 5.1. While there exists a number of datasets to evaluate single person pose estimation methods in videos, such as *e.g.*, J-HMDB (Jhuang et al., 2013) and Penn-Action (Zhang et al., 2013), none of the video datasets provides annotations to benchmark multi-person pose estimation and tracking at the same time. To allow for a quantitative evaluation of this problem, we therefore also introduce a new "Multi-Person PoseTrack" dataset which provides pose annotations for multiple persons in each video to measure pose estimation accuracy, and also provides a unique ID for each of the annotated persons to benchmark multi-person pose tracking, as shown in Figure 5.2 The proposed dataset introduces new challenges to the field of human pose estimation and tracking since it contains a large amount of appearance and pose variations, body part occlusion and truncation, large scale variations, fast camera and person movements, motion blur, and a sufficiently large number of persons per video. In order to evaluate the pose estimation and tracking accuracy, we introduce a new protocol that also deals with occluded body joints. We quantify the proposed method in detail on the proposed dataset, and also report results for several baseline methods. The source code, pre-trained models and the dataset are publicly available.[1]

5.2. MULTI-PERSON POSE TRACKING

Our method jointly solves the problem of multi-person pose estimation and tracking for all persons appearing in a video together. We first generate a set of joint detection candidates in each video as illustrated in Figure 5.3. From the detections, we build a graph consisting of spatial edges connecting the detections within a frame and temporal edges connecting detections of the same joint type

[1] http://pages.iai.uni-bonn.de/iqbal_umar/PoseTrack/

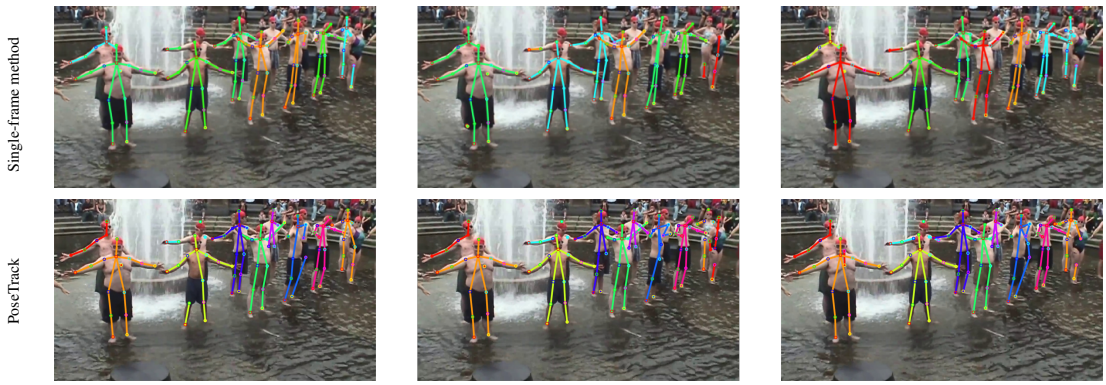


Figure 5.1: Comparison of the approach proposed in this chapter for multi-person pose estimation and tracking with the single-frame method presented in Chapter 4. Same color represents same person across frames. Note the color differences between pose estimates of single-frame based method. The approach presented in this chapter, on the other hand, also tracks the persons overtime.

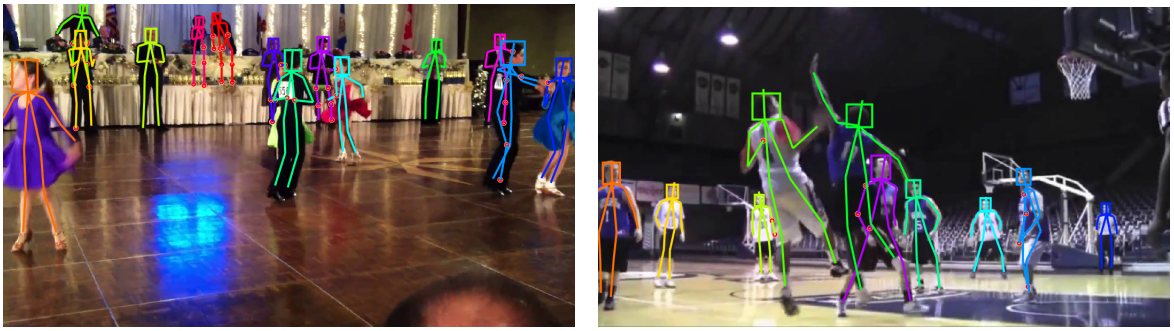


Figure 5.2: Example frames and annotations from the proposed Multi-Person PoseTrack dataset.

over frames. We solve the problem using integer linear programming (ILP) whose feasible solution provides the pose estimate for each person in all video frames, and also performs person association across frames. We first introduce the proposed method and discuss the proposed dataset for evaluation in Section 5.3.

5.2.1. Spatio-Temporal Graph

Given a video sequence $\mathcal{F}$ containing an arbitrary number of persons, we generate a set of body joint detection candidates $D = \{D_f\}_{f \in \mathcal{F}}$ where D_f is the set for frame f. Every detection $d \in D$ at location $\mathbf{x}_d^f \in \mathbb{R}^2$ in frame f belongs to a joint type $j \in \mathcal{J} = \{1, \ldots, J\}$. Additional details regarding the used detector will be provided in Section 5.2.4.

For multi-person pose tracking, we aim to identify the joint hypotheses that belong to an individual person in the entire video. This can be formulated by a graph structure $\mathcal{G} = (D, \mathcal{E})$ where D is the set of nodes. The set of edges $\mathcal{E}$ consists of two types of edges, namely spatial edges $\mathcal{E}_s$ and temporal edges $\mathcal{E}_t$. The spatial edges correspond to the union of edges of a fully connected graph for each frame, *i.e.*

$$\mathcal{E}_s = \bigcup_{f \in \mathcal{F}} \mathcal{E}_s^f \text{ and } \mathcal{E}_s^f = \{(d, d') : d \neq d' \wedge \ d, d' \in D_f\}. \tag{5.1}$$

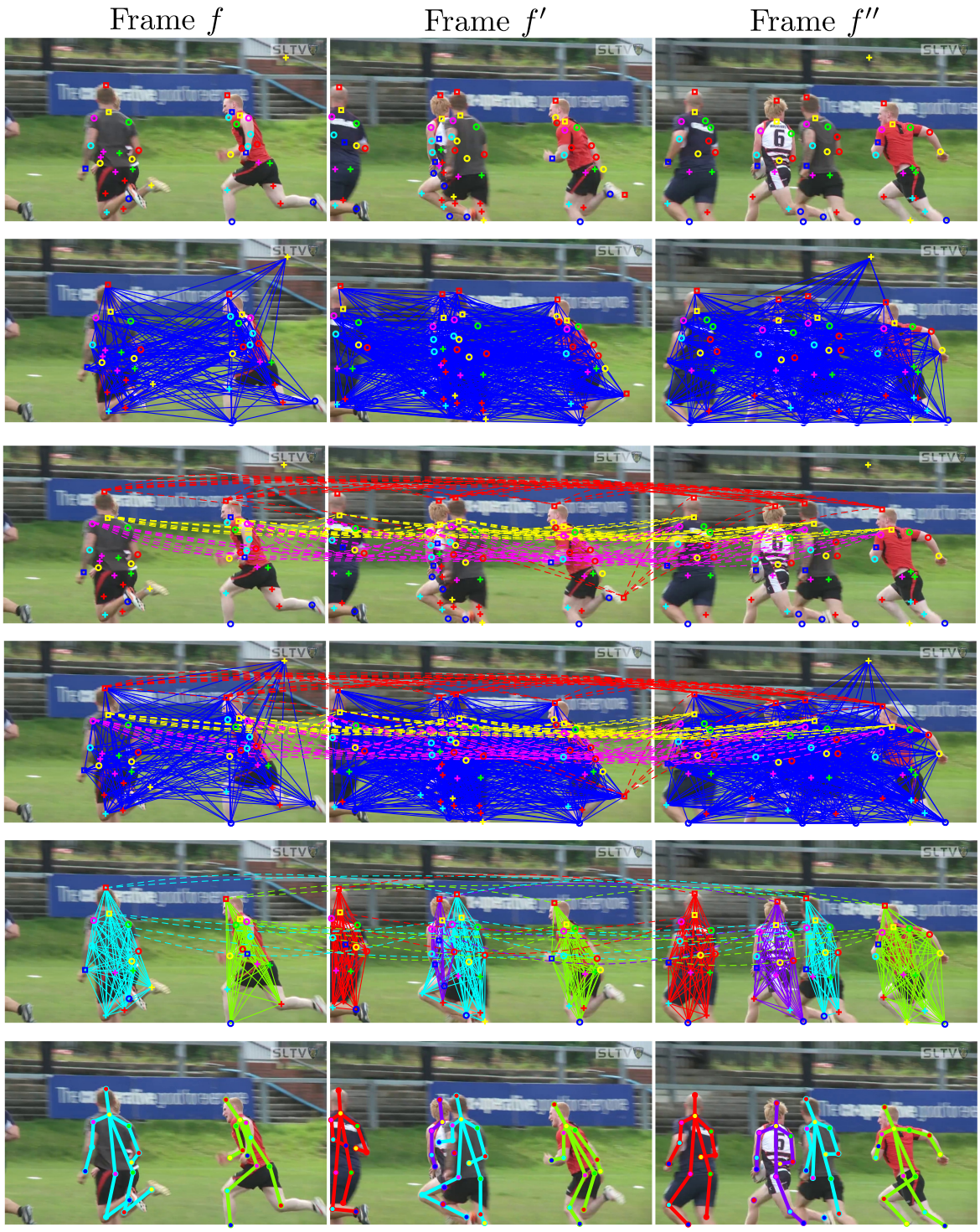


Figure 5.3: **Row-1:** Body joint detection hypotheses shown for three frames. **Row-2:** Spatial graph with edges between all detection candidates. **Row-3:** Temporal graph with temporal edges for head (red) and neck (yellow). **Row-4:** Spatio-temporal graph is constructed as the union of spatial graph and temporal graph. We only show a subset of the edges. **Row-5:** Connected components obtained after optimizing the spatio-temporal graph. Each color corresponds to a unique person identity. **Row-6:** Estimated poses for all persons in the video.

Note that these edges connect joint candidates independently of the associated joint type j. The temporal edges connect only joint hypotheses of the same joint type over two different frames, *i.e.*

$$\mathcal{E}_t = \{(d, d') : j{=}j' \wedge\ d \in D_f \wedge\ d' \in D_{f'} \\ \wedge\ 1{\leqslant}|f - f'|{\leqslant}\tau \wedge\ f, f' \in \mathcal{F}\}. \tag{5.2}$$

The temporal connections are not only modeled for neighboring frames, *i.e.*$|f - f'| = 1$, but we also take temporal relations up to τ frames into account to handle short-term occlusion and missing detections. The graph structure is illustrated in Figure 5.3.

5.2.2. Graph Partitioning

By removing edges and nodes from the graph $\mathcal{G} = (D, \mathcal{E})$, we obtain several partitions of the spatio-temporal graph and each partition corresponds to a tracked pose of an individual person. In order to solve the graph partitioning problem, we introduce the three binary vectors $v \in \{0,1\}^{|D|}$, $s \in \{0,1\}^{|\mathcal{E}_s|}$, and $t \in \{0,1\}^{|\mathcal{E}_t|}$. Each binary variable implies if a node or edge is removed, *i.e.*$v_d{=}0$ implies that the joint detection d is removed. Similarly, $s_{(d_f, d'_f)}{=}0$ with $(d_f, d'_f) \in \mathcal{E}_s$ implies that the spatial edge between the joint hypothesis d and d' in frame f is removed while $t_{(d_f, d'_{f'})}{=}0$ with $(d_f, d'_{f'}) \in \mathcal{E}_t$ implies that the temporal edge between the joint hypothesis d in frame f and d' in frame f' is removed.

A partitioning is obtained by minimizing the cost function

$$\arg\min_{v,s,t} \left(\langle v, \phi\rangle + \langle s, \psi_s\rangle + \langle t, \psi_t\rangle\right) \tag{5.3}$$

$$\langle v, \phi\rangle = \sum_{d \in D} v_d \phi(d) \tag{5.4}$$

$$\langle s, \psi_s\rangle = \sum_{(d_f, d'_f) \in \mathcal{E}_s} s_{(d_f, d'_f)} \psi_s(d_f, d'_f) \tag{5.5}$$

$$\langle t, \psi_t\rangle = \sum_{(d_f, d'_{f'}) \in \mathcal{E}_t} t_{(d_f, d'_{f'})} \psi_t(d_f, d'_{f'}). \tag{5.6}$$

This means that we search for a graph partitioning such that the cost of the remaining nodes and edges is minimal. The cost for a node d is defined by the unary term:

$$\phi(d) = \log \frac{1 - p_d}{p_d} \tag{5.7}$$

where $p_d \in (0, 1)$ corresponds to the probability of the joint hypothesis d. Note that $\phi(d)$ is negative when $p_d{>}0.5$ and detections with a high confidence are preferred since they reduce the cost function (5.3). The cost for a spatial or temporal edge is defined similarly by

$$\psi_s(d_f, d'_f) = \log \frac{1 - p^s_{(d_f, d'_f)}}{p^s_{(d_f, d'_f)}} \tag{5.8}$$

$$\psi_t(d_f, d'_{f'}) = \log \frac{1 - p^t_{(d_f, d'_{f'})}}{p^t_{(d_f, d'_{f'})}}. \tag{5.9}$$

While p^s denotes the probability that two joint detections d and d' in a frame f belong to the same person, p^t denotes the probability that two detections of a joint in frame f and f' are the same. In Section 5.2.4 we will discuss how the probabilities p_d, $p^s_{(d_f,d'_f)}$, and $p^t_{(d_f,d'_{f'})}$ are learned.

In order to ensure that the feasible solutions of the objective (5.3) result in well defined body poses and valid pose tracks, we have to add additional constraints. The first set of constraints ensures that two joint hypotheses are associated to the same person ($s_{(d_f,d'_f)}$=1) only if both detections are considered as valid, *i.e.*, v_{d_f}=1 and $v_{d'_f}$=1:

$$s_{(d_f,d'_f)} \leqslant v_{d_f} \ \wedge \ s_{(d_f,d'_f)} \leqslant v_{d'_f} \quad \forall (d_f, d'_f) \in \mathcal{E}_s. \tag{5.10}$$

The same holds for the temporal edges:

$$t_{(d_f,d'_{f'})} \leqslant v_{d_f} \ \wedge \ t_{(d_f,d'_{f'})} \leqslant v_{d'_{f'}} \quad \forall (d_f, d'_{f'}) \in \mathcal{E}_t. \tag{5.11}$$

The second set of constraints are transitivity constraints in the spatial domain. Such transitivity constraints have been discussed before in Chapter 4 for multi-person pose estimation in images. They enforce for any triplet of joint detection candidates (d_f, d'_f, d''_f) that if d_f and d'_f are associated to one person and d'_f and d''_f are also associated to one person, *i.e.*$s_{(d_f,d'_f)}$=1 and $s_{(d'_f,d''_f)}$=1, then the edge (d_f, d''_f) should also be added:

$$\begin{aligned} s_{(d_f,d'_f)} + s_{(d'_f,d''_f)} - 1 \leqslant s_{(d_f,d''_f)} \\ \forall (d_f, d'_f), (d'_f, d''_f) \in \mathcal{E}_s. \end{aligned} \tag{5.12}$$

An example of a transitivity constraint is illustrated in Figure 5.4a. The transitivity constraints can be used to enforce that a human can have only one joint type j, *e.g.*, only one head. Let d_f and d''_f have the same joint type j while d'_f belongs to another joint type j'. Without transitivity constraints connecting d_f and d''_f with d'_f might result in a low cost. The transitivity constraints, however, enforce that the binary cost $\psi_s(d_f, d''_f)$ is added. To prevent poses with multiple joints, we thus only have to ensure that the binary cost $\psi_s(d, d'')$ is very high if j=j''. We discuss this more in detail in Section 5.2.4.

In contrast to previous work, we also have to ensure spatio-temporal consistency. Similar to the spatial transitivity constraints (5.12), we can define temporal transitivity constraints:

$$\begin{aligned} t_{(d_f,d'_{f'})} + t_{(d'_{f'},d''_{f''})} - 1 \leqslant t_{(d_f,d''_{f''})} \\ \forall (d_f, d'_{f'}), (d'_{f'}, d''_{f''}) \in \mathcal{E}_t. \end{aligned} \tag{5.13}$$

The last set of constraints are spatio-temporal constraints that ensure that the pose is consistent over time. We define two types of spatio-temporal constraints. The first type consists of a triplet of joint detection candidates $(d_f, d'_{f'}, d''_{f'})$ from two different frames f and f'. The constraints are defined as,

$$\begin{aligned} t_{(d_f,d'_{f'})} + t_{(d_f,d''_{f'})} - 1 \leqslant s_{(d'_{f'},d''_{f'})} \\ t_{(d_f,d'_{f'})} + s_{(d'_{f'},d''_{f'})} - 1 \leqslant t_{(d_f,d''_{f'})} \\ \forall (d_f, d'_{f'}), (d_f, d''_{f'}) \in \mathcal{E}_t, \end{aligned} \tag{5.14}$$

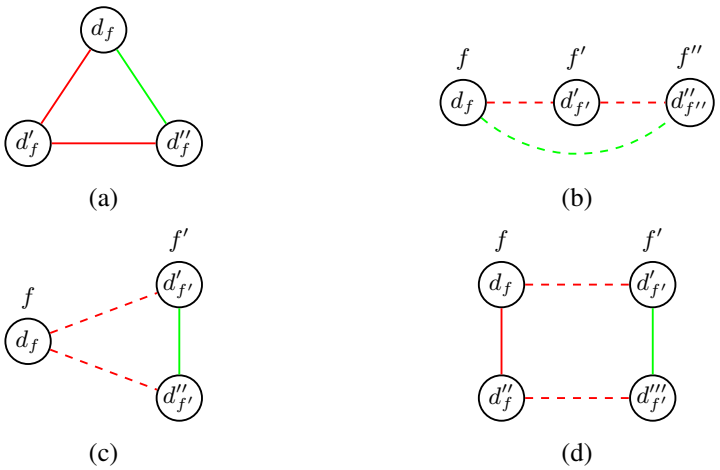


Figure 5.4: *(a)* The spatial transitivity constraints (5.12) ensure that if the two joint hypotheses d_f and d''_f are spatially connected to d'_f (red edges) then the cost of the spatial edge between d_f and d''_f (green edge) also has to be added. *(b)* The temporal transitivity constraints (5.13) ensure transitivity for temporal edges (dashed). *(c)* The spatio-temporal transitivity constraints (5.14) model transitivity for two temporal edges and one spatial edge. *(d)* The spatio-temporal consistency constraints (5.15) ensure that if two pairs of joint hypotheses $(d_f, d'_{f'})$ and $(d''_f, d'''_{f'})$ are temporally connected (dashed red edges) and d_f and d''_f are spatially connected (solid red edge) then the cost of the spatial edge between $d'_{f'}$ and $d'''_{f'}$ (solid green edge) also has to be added.

and enforce transitivity for two temporal edges and one spatial edge. The second type of spatio-temporal constraints are based on quadruples of joint detection candidates $(d_f, d'_{f'}, d''_f, d'''_{f'})$ from two different frames f and f'. The spatio-temporal constraints ensure that if $(d_f, d'_{f'})$ and $(d''_f, d'''_{f'})$ are temporally connected and (d_f, d''_f) are spatially connected then the spatial edge $(d'_{f'}, d'''_{f'})$ has to be added:

$$\begin{aligned} t_{(d_f, d'_{f'})} + t_{(d''_f, d'''_{f'})} + s_{(d_f, d''_f)} - 2 \leqslant s_{(d'_{f'}, d'''_{f'})} \\ t_{(d_f, d'_{f'})} + t_{(d''_f, d'''_{f'})} + s_{(d'_{f'}, d'''_{f'})} - 2 \leqslant s_{(d_f, d''_f)} \\ \forall (d_f, d'_{f'}), (d''_f, d'''_{f'}) \in \mathcal{E}_t. \end{aligned} \tag{5.15}$$

An example of both types of spatio-temporal constraint can be seen in Figure 5.4c and Figure 5.4d, respectively.

5.2.3. Optimization

We optimize the objective (5.3) with the branch-and-cut algorithm of the ILP solver Gurobi. To reduce the runtime for long sequences, we process the video batch-wise where each batch consists of $k = 31$ frames. For the first k frames, we build the spatio-temporal graph as discussed and optimize the objective (5.3). We then continue to build a graph for the next k frames and add the previously selected nodes and edges to the graph, but fix them such that they cannot be removed anymore. Since the graph partitioning produces also small partitions, which usually correspond to clusters of false positive joint detections, we remove any partition that is shorter than 7 frames or has less than 6 nodes per frame on average.

Figure 5.5: Example of the dense correspondences used for temporal potentials. The top row shows the joint detection candidates with the defined bounding boxes to extract the feature vectors for temporal association. The bottom row shows the correspondences found by the DeepMatching (Weinzaepfel et al., 2013) algorithm.

5.2.4. Potentials

In order to compute the unaries ϕ (5.7) and binaries ψ (5.8),(5.9), we have to learn the probabilities p_d, $p^s_{(d_f,d'_f)}$, and $p^t_{(d_f,d'_{f'})}$.

The probability p_d is given by the confidence of the joint detector. As joint detector, we use the publicly available pre-trained CNN (Insafutdinov et al., 2016) trained on the MPII Multi-Person Pose dataset (Pishchulin et al., 2016). In contrast to (Insafutdinov et al., 2016), we do not assume that any scale information is given. We therefore apply the detector to an image pyramid with 4 scales $\gamma \in \{0.6, 0.9, 1.2, 1.5\}$. For each detection d located at $\mathbf{x}^f_d$, we compute a quadratic bounding box $B_d = \{\mathbf{x}^f_d, h_d\}$. We use $h_d = \frac{70}{\gamma}$ for the width and height. To reduce the number of detections, we remove all bounding boxes that have an intersection-over-union (IoU) ratio over 0.7 with another bounding box that has a higher detection confidence.

The spatial probability $p^s_{(d_f,d'_f)}$ depends on the joint types j and j' of the detections. If $j{=}j'$, we define $p^s_{(d_f,d'_f)}{=}\text{IoU}(B_d, B_{d'})$. This means that a joint type j cannot be added multiple times to a person except if the detections are very close. If a partition includes detections of the same type in a single frame, the detections are merged by computing the weighted mean of the detections, where the weights are proportional to p_d. If $j{\neq}j'$, we use the pre-trained binaries (Insafutdinov et al., 2016) after a scale normalization.

The temporal probability $p^t_{(d_f,d'_{f'})}$ should be high if two detections of the same joint type at different frames belong to the same person. To that end, we build on the idea recently used in multi-person tracking (Tang et al., 2016) and compute dense correspondences between two frames using Deep-

Dataset	Video-labeled poses	multi-person	Large scale variation	variable skeleton size	# of persons
Leeds Sports (Johnson and Everingham, 2010b)					2000
MPII Pose (Andriluka et al., 2014)			✓	✓	40,522
We Are Family (Eichner and Ferrari, 2010)		✓			3131
MPII Multi-Person Pose (Pishchulin et al., 2016)		✓	✓	✓	14,161
MS-COCO Keypoints (Lin et al., 2014a)		✓	✓	✓	250,000
AIChallenger (Wu et al., 2017)		✓	✓	✓	700,000
J-HMDB (Jhuang et al., 2013)	✓		✓	✓	32,173
Penn-Action (Zhang et al., 2013)	✓		✓		159,633
VideoPose (Sapp et al., 2011)	✓				1286
Poses-in-the-wild (Cherian et al., 2014)	✓				831
YouTube Pose (Charles et al., 2016)	✓				5000
FYDP (Shen et al., 2014)	✓				1680
UYDP (Shen et al., 2014)	✓				2000
Multi-Person PoseTrack	✓	✓	✓	✓	16,219

Table 5.1: A comparison of PoseTrack dataset with the existing related datasets for human pose estimation in images and videos.

Matching (Weinzaepfel et al., 2013). Let K_{d_f} and $K_{d'_{f'}}$ be the sets of matched key-points inside the bounding boxes B_{d_f} and $B_{d'_{f'}}$ and $\underline{K}_{dd'}=|K_{d_f} \cup K_{d'_{f'}}|$ and $\overline{K}_{dd'}=|K_{d_f} \cap K_{d'_{f'}}|$ the union and intersection of these two sets. We then form a feature vector by $\{\overline{K}/\underline{K}, \min(p_d, p_{d'}), \Delta\mathbf{x}_{dd'}, \|\Delta\mathbf{x}_{dd'}\|\}$ where $\Delta\mathbf{x}_{dd'} = \mathbf{x}_d^f - \mathbf{x}_{d'}^{f'}$. We also append the feature vector with non-linear terms as done in (Tang et al., 2016). The mapping from the feature vector to the probability $p^t_{(d_f, d'_{f'})}$ is obtained by logistic regression. An example of dense correspondences between two frames can be seen in Fig 5.5.

5.3. THE MULTI-PERSON POSETRACK DATASET

In this section we introduce our new dataset for multi-person pose estimation in videos. The MPII Multi-Person Pose (Andriluka et al., 2014) is currently one of the most popular benchmarks for multi-person pose estimation in images, and covers a wide range of activities. For each annotated image, the dataset also provides unlabeled video clips ranging 20 frames both forward and backward in time relative to that image. For our video dataset, we manually select a subset of all available videos that contain multiple persons and cover a wide variety of person-person or person-object interactions. Moreover, the selected videos are chosen to contain a large amount of body pose appearance and scale variation, as well as body part occlusion and truncation. The videos also contain severe body motion, *i.e.*, people occlude each other, re-appear after complete occlusion, vary in scale across the video, and also significantly change their body pose. The number of visible persons and body parts may also vary during the video. The duration of all provided video clips is exactly 41 frames. To include longer and variable-length sequences, we downloaded the original raw video clips using the provided URLs and obtained an additional set of videos. To prevent an overlap with the existing data, we only considered sequences that are at least 150 frames apart from the training samples, and followed the same rationale as above to ensure diversity.

In total, we compiled a set of 60 videos with the number of frames per video ranging between 41

and 151. The number of persons ranges between 2 and 16 with an average of more than 5 persons per video sequence, totaling over 16,000 annotated poses. The person heights are between 100 and 1200 pixels. We split the dataset into a training and testing set with an equal number of videos.

5.3.1. Annotation

As in (Andriluka et al., 2014), we annotate 14 body joints and a rectangle enclosing the person's head. The latter is required to estimate the absolute scale which is used for evaluation. We assign a unique identity to every person appearing in the video. This person ID remains the same throughout the video until the person moves out of the field-of-view. Since we do not target person re-identification in this work, we assign a new ID if a person re-appears in the video. We also provide occlusion flags for all body joints. A joint is marked occluded if it was in the field-of-view but became invisible due to an occlusion. Truncated joints, *i.e.*, those outside the image border limits, are not annotated, therefore, the number of joints per person varies across the dataset. Very small persons were zoomed in to a reasonable size to accurately perform the annotation. To ensure a high quality of the annotation, all annotations were performed by trained in-house workers, following a clearly defined protocol. An example annotation can be seen in Figure 5.2.

5.3.2. Experimental setup and evaluation metrics

Since the problem of simultaneous multi-person pose estimation and person tracking has not been quantitatively evaluated in the literature, we define a new evaluation protocol for this problem. To this end, 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 part is predicted correctly, we use the widely adopted 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 the 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, Andriluka et al. (2014) propose to use 30% of the head box diagonal. We have found this threshold to be too relaxed because recent pose estimation approaches are capable of predicting the joint locations rather accurately. Therefore, we use a more strict evaluation with a 20% threshold.

Given the joint localization threshold for each person, we compute two sets of evaluation metrics, one adopted from the multi-target tracking literature (Yang and Nevatia, 2012; Choi, 2015; Milan et al., 2016) to evaluate multi-person pose tracking, and one which is commonly used for evaluating multi-person pose estimation (Pishchulin et al., 2016).

Tracking. To evaluate multi-person pose tracking, we consider each joint trajectory as one individual target,[2] and compute multiple measures. First, the CLEAR MOT metrics (Bernardin and Stiefelhagen, 2008) provide the tracking accuracy (MOTA) and tracking precision (MOTP). The former is derived from three types of error ratios: false positives, missed targets, and identity switches (IDs). These are linearly combined to produce a normalized accuracy where 100% corresponds to zero errors. MOTP measures how precise each object, or in our case each body joint, has been localized w.r.t the ground-truth. Second, we report trajectory-based measures proposed in (Li et al., 2009),

[2]Note that only joints of the same type are matched.

that count the number of mostly tracked (MT) and mostly lost (ML) tracks. A track is considered mostly tracked if it has been recovered in at least 80% of its length, and mostly lost if more than 80% are not tracked. For completeness, we also compute the number of times a ground-truth trajectory is fragmented (FM).

Pose. For measuring frame-wise multi-person pose accuracy, we use *Mean Average Precision* (mAP) as is done in (Pishchulin et al., 2016). The protocol to evaluate multi-person pose estimation in (Pishchulin et al., 2016) assumes that the rough scale and location of a group of persons is known during testing (Pishchulin et al., 2016), which is not the case in realistic scenarios, and in particular in videos. We therefore propose to make no assumption during testing and evaluate the predictions without rescaling or shifting them according to the ground-truth.

Occlusion handling. Both of the aforementioned protocols to measure pose estimation and tracking accuracy do not consider occlusion during evaluation, and penalize if an occluded target that is annotated in the ground-truth is not correctly estimated (Milan et al., 2016; Pishchulin et al., 2016). This, however, discourages methods that either detect occlusion and do not predict the occluded joints or approaches that predict the joint position even for occluded joints. We want to provide a fair comparison for both types of occlusion handling. We therefore extend both measures to incorporate occlusion information explicitly. To this end, we first assign each person to one of the ground-truth poses based on the PCKh measure as done in (Pishchulin et al., 2016). For each matched person, we consider an occluded joint correctly estimated either if *a)* it is predicted at the correct location despite being occluded, or *b)* it is not predicted at all. Otherwise, the prediction is considered as a false positive.

5.4. EXPERIMENTS

In this section we evaluate the proposed method for joint multi-person pose estimation and tracking on the newly introduced Multi-Person PoseTrack dataset.

5.4.1. Multi-Person Pose Tracking

The results for multi-person pose tracking (MOT CLEAR metrics) are reported in Table 5.2. To find the best setting, we first perform a series of experiments, investigating the influence of temporal connection density, temporal connection length, and inclusion of different constraint types.

We first examine the impact of different joint combinations for temporal connections. Connecting only the Head Tops (HT) between frames results in a Multi-Object Tracking Accuracy (MOTA) of 27.2 with a recall and precision of 57.6% and 66.0%, respectively. Adding Neck and Shoulder (HT:N:S) detections for temporal connections improves the MOTA score to 28.2, while also improving the recall from 57.6% to 62.7%. Adding more temporal connections also increases other metrics such as MT, ML, and also results in a lower number of ID switches (IDs) and fragments (FM). However, increasing the number of joints for temporal edges even further (HT:N:S:H) results in a slight decrease in performance. This is most likely due to the weaker DeepMatching correspondences between hip joints, which are difficult to match. When only the body extremities (HT:W:A) are used for temporal edges, we obtain a similar MOTA as for (HT:N:S), but slightly worse other tracking measures. Considering the MOTA performance and the complexity of our graph structure, we use (HT:N:S) as our default setting.

Method	Rcll ↑	Prcn ↑	MT ↑	ML ↓	IDs ↓	FM ↓	MOTA ↑	MOTP ↑
Impact of temporal connection density								
HT	57.6	66.0	632	623	674	5080	27.2	56.1
HT:N:S	62.7	64.9	760	510	470	5557	28.2	55.8
HT:N:S:H	63.1	64.5	774	494	478	5564	27.8	55.7
HT:W:A	62.8	64.9	758	526	516	5458	28.2	55.8
Impact of the length of temporal connection (τ)								
HT:N:S ($\tau = 1$)	62.7	64.9	760	510	470	5557	28.2	55.8
HT:N:S ($\tau = 3$)	63.0	64.8	775	502	431	5629	28.2	55.7
HT:N:S ($\tau = 5$)	62.8	64.7	763	508	381	5676	28.0	55.7
Impact of the constraints								
All	63.0	64.8	775	502	431	5629	28.2	55.7
All \ spat. transitivity	22.2	76.0	115	1521	39	3947	15.1	58.0
All \ temp. transitivity	60.3	65.1	712	544	268	5610	27.7	55.8
All \ spatio-temporal	55.1	64.1	592	628	262	5444	23.9	55.7
Comparison with the Baselines								
Ours	**63.0**	**64.8**	**775**	**502**	431	5629	**28.2**	**55.7**
BBox-Tracking (Tang et al., 2016; Ren et al., 2015)								
+ LJPA (Chapter 4)	58.8	64.8	716	646	**319**	**5026**	26.6	53.5
+ CPM (Wei et al., 2016)	60.1	57.7	754	611	347	4969	15.6	53.4

Table 5.2: Quantitative evaluation of multi-person pose-tracking using common multi-object tracking metrics. Up and down arrows indicate whether higher or lower values for each metric are better. The first three blocks of the table present an ablative study on design choices w.r.tjoint selection, temporal edges, and constraints. The bottom part compares our final result with two strong baselines described in the text. HT:Head Top, N:Neck, S:Shoulders, W:Wrists, A:Ankles

Instead of considering only neighboring frames for temporal edges, we also evaluate the tracking performance while introducing longer-range temporal edges of up to 3 and 5 frames. Adding temporal edges between detections that are at most three frames ($\tau = 3$) apart improves the performance only slightly, whereas increasing the distance even further ($\tau = 5$) worsens the performance. For the rest of our experiments we therefore set $\tau = 3$.

To evaluate the proposed optimization objective (5.3) for joint multi-person pose estimation and tracking in more detail, we have quantified the impact of various kinds of constraints (5.10)-(5.15) enforced during the optimization. To this end, we remove one type of constraints at a time and solve the optimization problem. As shown in Table 5.2, all types of constraints are important to achieve best performance, with the spatial transitivity constraints playing the most crucial role. This is expected since these constraints ensure that we obtain valid poses without multiple joint types assigned to one person. Temporal transitivity and spatio-temporal constraints also turn out to be important to obtain good results. Removing either of the two significantly decreases the recall, resulting in a drop in MOTA.

Since we are the first to report results on the Multi-Person PoseTrack dataset, we also develop two baseline methods by using the existing approaches. For this, we rely on our approach in Chapter 4 for multi-person pose estimation in images. As discussed in Chapter 4, we use a person detector (Ren et al., 2015) to first obtain person bounding box hypotheses, and then estimates the pose for each person independently. We extend our approach to videos as follows. We first generate person bounding boxes for all frames in the video using a state-of-the-art person detector (Faster R-CNN (Ren et al., 2015)), and perform person tracking using a state-of-the-art person tracker (Tang et al., 2016) and train it on the training set of the Multi-Person PoseTrack Dataset. We also discard all tracks that are shorter than 7 frames. The final pose estimates are obtained by using the Local Joint-to-Person Association (LJPA) approach presented in Chapter 4 for each person track. We also report results when Convolutional Pose Machines (CPM) (Wei et al., 2016) are used instead. Since CPM does not account for joint occlusion and truncation, the MOTA score is significantly lower than for LJPA. LJPA (Chapter 4) improves the performance, but remains inferior w.r.t most measures compared to our proposed method. In particular, our method achieves the highest MOTA and MOTP scores. The former is due to a significantly higher recall, while the latter is a result of a more precise part localization. Interestingly, the person bounding-box tracking based baselines achieve a lower number of ID switches. We believe that this is primarily due to the powerful multi-target tracking approach (Tang et al., 2016), which can handle person identities more robustly.

5.4.2. Frame-wise Multi-Person Pose Estimation

The results for frame-wise multi-person pose estimation (mAP) are summarized in Table 5.3. Similar to the evaluation for pose tracking, we evaluate the impact of spatio-temporal connection density, length of temporal connections and the influence of different constraint types. Having connections only between Head Top (HT) detections results in a mAP of 34.3%. As for pose tracking, introducing temporal connections for Neck and Shoulders (HT:N:S) results in a higher accuracy and improves the mAP from 34.3% to 37.9%. The mAP elevates slightly more when we also incorporate connections for hip joints (HT:N:S:H). This is in contrast to pose tracking where MOTA dropped slightly when we also use connections for hip joints. As before, inclusion of edges between all detections that are in the range of 3 frames improves the performance, while increasing the distance further ($\tau = 5$)

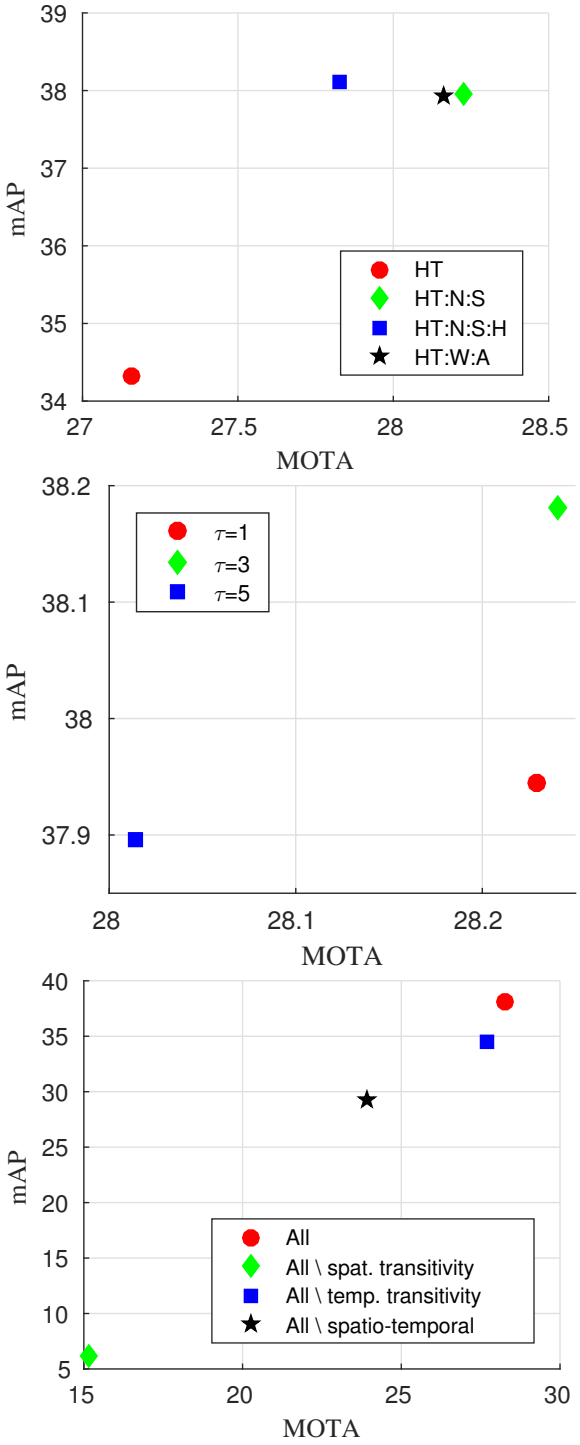


Figure 5.6: **Top** Impact of the the temporal edge density. **Middle** Impact of the length of temporal edges. **Bottom** Impact of different constraint types.

Method	Head	Sho	Elb	Wri	Hip	Knee	Ank	mAP
Impact of the temporal connection density								
HT	52.5	47.0	37.6	28.2	19.7	27.8	27.4	34.3
HT:N:S	56.1	51.3	42.1	31.2	22.0	31.6	31.3	37.9
HT:N:S:H	56.3	51.5	42.2	31.4	21.7	31.6	32.0	38.1
HT:W:A	56.0	51.2	42.2	31.6	21.6	31.2	31.7	37.9
Impact of the length of temporal connection (τ)								
HT:N:S ($\tau = 1$)	56.1	51.3	42.1	31.2	22.0	31.6	31.3	37.9
HT:N:S ($\tau = 3$)	56.5	51.6	42.3	31.4	22.0	31.9	31.6	38.2
HT:N:S ($\tau = 5$)	56.2	51.3	41.8	31.1	22.0	31.4	31.5	37.9
Impact of the constraints								
All	56.5	51.6	42.3	31.4	22.0	31.9	31.6	38.2
All \ spat. transitivity	7.8	10.1	7.2	4.6	2.7	4.9	5.9	6.2
All \ temp. transitivity	50.5	46.8	37.5	27.6	20.3	30.1	28.7	34.5
All \ spatio-temporal	42.3	40.8	32.8	24.3	17.0	25.3	22.4	29.3
Comparison with the state-of-the-art								
Ours	**56.5**	51.6	**42.3**	**31.4**	22.0	**31.9**	**31.6**	**38.2**
BBox-Detection (Ren et al., 2015)								
+ LJPA (Chapter 4)	50.5	49.3	38.3	33.0	21.7	29.6	29.2	35.9
+ CPM (Wei et al., 2016)	48.8	47.5	35.8	29.2	20.7	27.1	22.4	33.1
DeeperCut (Insafutdinov et al., 2016)	56.2	**52.4**	40.1	30.0	**22.8**	30.5	30.8	37.5

Table 5.3: Quantitative evaluation of multi-person pose estimation (mAP). HT:Head Top, N:Neck, S:Shoulders, W:Wrists, A:Ankles

starts to deteriorate the performance. A similar trend can also been seen for the impact of different types of constraints. The removal of spatial transitivity constraints results in a drastic decrease in pose estimation accuracy. Without temporal transitivity constraints or spatio-temporal constraints the pose estimation accuracy drops by more than 3% and 8%, respectively. This once again indicates that all types of constraints are essential to obtain better pose estimation and tracking performance.

We also compare the proposed method with the competiting approaches for multi-person pose estimation in images. For LJPA (Chapter 4), we use exactly the same details as in in Chapter 4. For CPM (Wei et al., 2016), we use the publicaly availabe source code. We can see that person bounding box based approaches significantly underperform as compared to the proposed method. We also compare with the state-of-the-art method DeeperCut (Insafutdinov et al., 2016). The approach, however, requires the rough scale of the persons during testing. For this, we use the person detections obtained from (Ren et al., 2015) to compute the scale using the median scale of all detected persons.

Our approach achieves a better performance than all other methods. Moreover, all these approaches require an additional person detector either to get the bounding boxes (LJPA and CPM), or the rough scale of the persons (Insafutdinov et al., 2016). Our approach on the other hand does not require a separate person detector, and we perform joint detection across different scales, while also solving the person association problem across frames.

We also visualize how multi-person pose estimation accuracy (mAP) relates with the multi-person tracking accuracy (MOTA) in Figure 5.6. Table 5.4 provides mean and median runtimes for constructing and solving the spatio-temporal graph along with the graph size for $k = 31$ frames over all test videos. Finally we provide some qualitative results in Figure 5.7-5.9. Some video results can be seen at `https://youtu.be/SgiFPWNuAGw`.

	Runtime (sec./frame)	# of nodes	# of spatial edges	# of temp. edges
Mean	14.7	2084	65535	12903
Median	4.2	1907	58164	8540

Table 5.4: Runtime and size of the spatio-temporal graph ($\tau = 3$, HT:N:S, $k = 31$), measured on a single threaded 3.3GHz CPU .

5.5. SUMMARY

In this chapter, we have presented a novel approach to simultaneously perform multi-person pose estimation and tracking. We demonstrate that the problem can be formulated as a spatio-temporal graph which can be efficiently optimized using integer linear programming. We have also presented a challenging and diverse annotated dataset with a comprehensive evaluation protocol to analyze the algorithms for multi-person pose estimation and tracking. Following the evaluation protocol, the proposed method does not make any assumptions about the number, size, or location of the persons, and can perform pose estimation and tracking in completely unconstrained videos. Moreover, the method is able to perform pose estimation and tracking under severe occlusion and truncation. Experimental results on the proposed dataset demonstrate that our method outperforms other baseline methods.

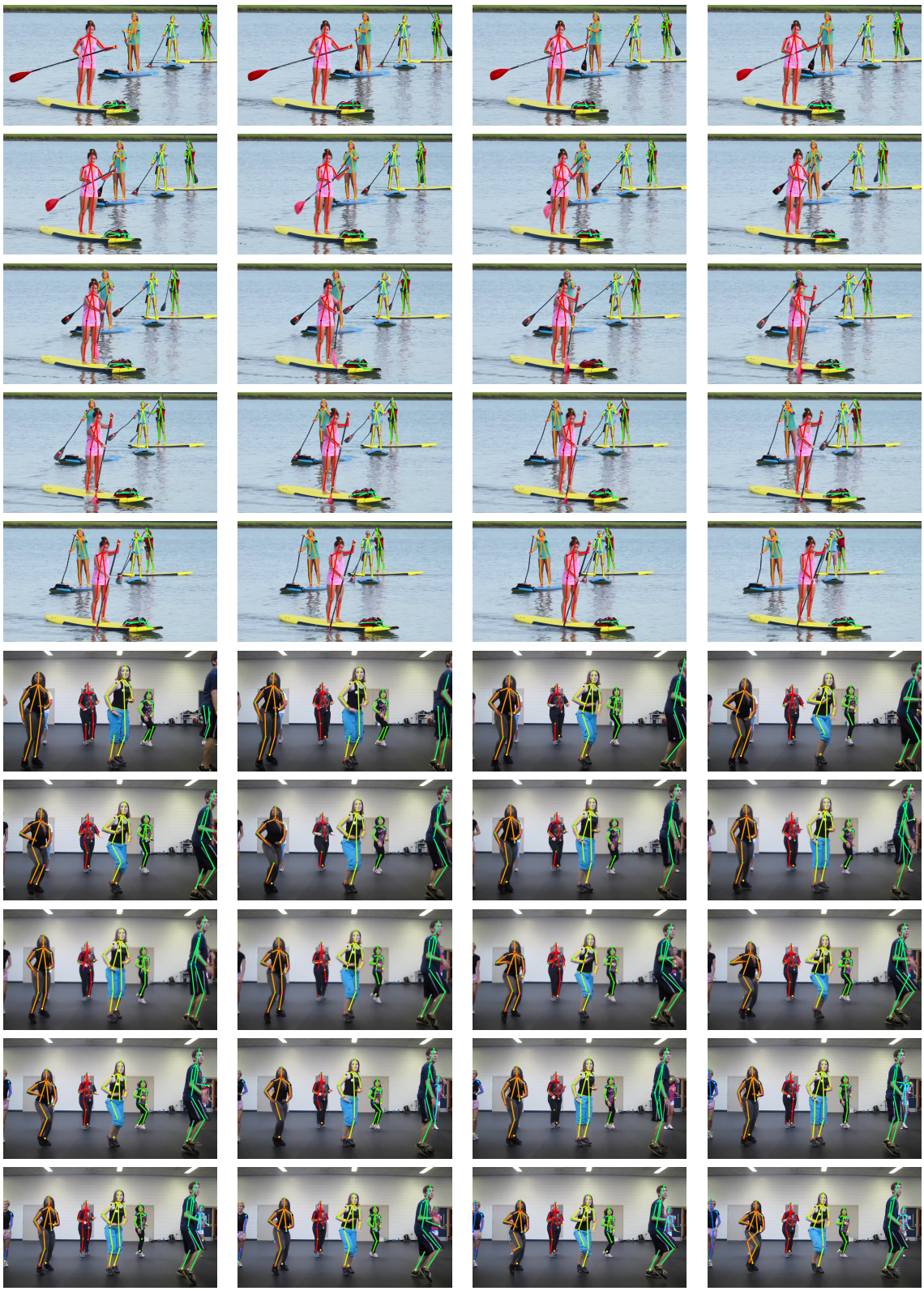


Figure 5.7: Qualitative Results. Visualization of the pose estimation and tracking results of our proposed approach on Multi-Person Pose-Track dataset. We show every second frame for each video clip. Our approach can estimate poses under severe occlusions and truncations. Note for example the orange person in the first video, and person appearings from the left/right sides in the second video.

Figure 5.8: Qualitative Results. Visualization of the pose estimation and tracking results of our proposed approach on Multi-Person Pose-Track dataset. We show every second frame for each video clip. Note that our approach can handle fast motion, motion blur and complex body articulations.

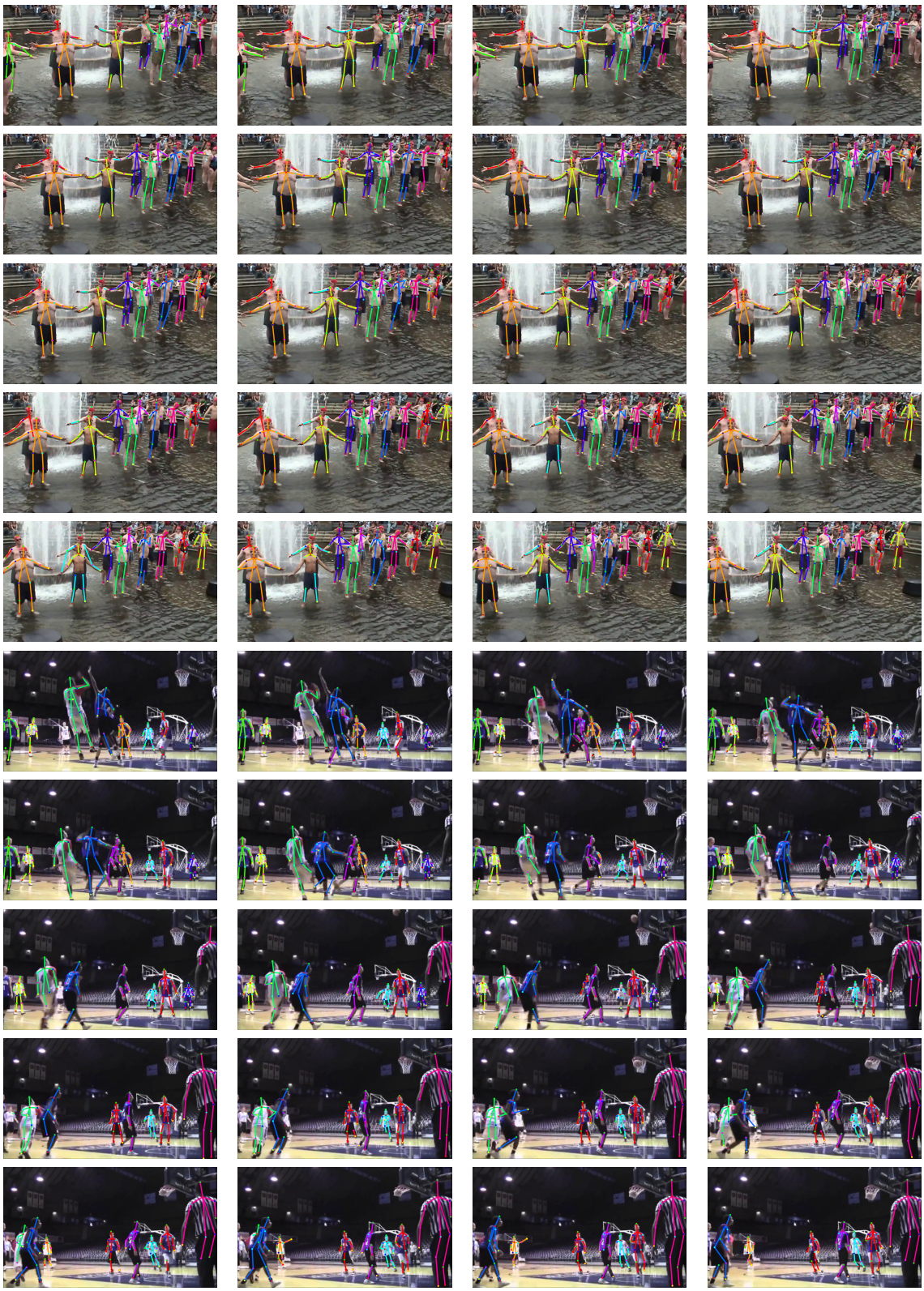


Figure 5.9: Qualitative Results. Visualization of the pose estimation and tracking results of our proposed approach on Multi-Person Pose-Track dataset. We show every second frame for each video clip. Note that our approach can estimate poses under severe occlusions and truncations, clutter, complex background and large scale variation.

CHAPTER 6

PoseTrack: A Benchmark for Human Pose Estimation and Tracking

In Chapter 5 we presented a new unconstrained dataset for multi-person pose estimation and tracking. We also proposed a comprehensive evaluation protocol to evaluate the proposed methods for this challenging problem quantitatively. While the proposed dataset provided a good starting point, it is very small to capture diverse real-world scenarios and to learn stronger pose estimation models.

To address these shortcomings, in this chapter we extend our dataset from Chapter 5 by a large margin and introduce PoseTrack which is a new large-scale benchmark for video-based human pose estimation and tracking. Furthermore, we conduct an extensive experimental study on recent methods for multi-person pose estimation and tracking and provide analysis of the strengths and weaknesses of the state-of-the-art.

Contents

6.1. INTRODUCTION

The significant progress in single-frame human pose estimation 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., 2014b). In Chapter 5, we provided a dataset for multi-person pose estimation in videos. However, as compared to the datasets available for multi-person pose estimation in images,

the dataset is at a very small scale and provides only 60 videos with most sequences containing only 41 frames. While the dataset makes a first step toward solving the problem at hand, it is certainly not enough to cover a large range of real-world scenarios and to learn stronger pose estimation models. In this chapter, we aim to fill this gap by extending the dataset proposed in Chapter 5 to a large-scale, high-quality benchmark for video-based multi-person pose estimation and tracking. To that end, we collect new videos, annotate them in an accurate and unified manner, and release a new large-scale dataset. Further, we also provide an online evaluation platform to ensure direct and fair performance comparison across numerous competing approaches.

Our new benchmark is organized around three related tasks focusing on single-frame multi-person pose estimation, multi-person pose estimation in video, and multi-person pose estimation and tracking. While the main focus of the dataset is still on multi-person pose estimation and tracking, progress in the single-frame setting will inevitably improve overall tracking quality. We thus make the single frame multi-person pose estimation 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. The proposed benchmark will be 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 a workshop and a competition at ICCV'17[1]. We obtained largely positive feedback from the twelve teams that participated in the competition. We incorporate some of this feedback into this work. In addition we analyze the currently best performing approaches and highlight the common difficulties for the problem of multi-person pose estimation and articulated tracking.

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/.

6.2. RELATED DATASETS

The commonly used publicly available datasets for evaluation of 2D human pose estimation are summarized in Table 6.1. 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, 2010a), "LSP Extended" (Johnson and Everingham, 2011), "MPII Human Pose (Single Person)" (Andriluka et al., 2014) and MS COCO Keypoints Challenge (Lin et al., 2014b). LSP and LSP Extended datasets focus on sports scenes featuring a few sport types. Although a combination of both datasets results in 1.1000×10^4 training poses, the evaluation set of 1.000×10^3 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 2.6000×10^4 annotated poses with 7.000×10^3 poses held out for evaluation. Both benchmarks focus on single person pose estimation and provide rough location scale of a person in question. In contrast, our dataset

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

Dataset	# of persons	Multi-person	Video-labeled poses	Data type
LSP (Johnson and Everingham, 2010a)	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
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., 2014b)	105,698	✓		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
FYDP (Shen et al., 2014)	✓	1680	✓	dance
UYDP (Shen et al., 2014)	✓	2000	✓	dance
Multi-Person PoseTrack (Chapter 5)	16,219	✓	✓	diverse
Proposed	**153,615**	✓	✓	**diverse**

Table 6.1: 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.

Figure 6.1: Example frames and annotations from 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 in (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., 2014b) has been introduced to provide a new large-scale benchmark for single frame based multi-person pose estimation. All these datasets are only limited to single-frame based body pose estimation. In contrast, our dataset also focuses on a 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 include about five seconds of video footage centered on 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.

Similar to our Multi-Person PoseTrack dataset in Chapter 5, recently, Insafutdinov et al. (2017) also provided a dataset for multi-person pose estimation in videos. However, similar to ours, it is also at a very small scale, and provides only 30 videos containing 20 frames each. While both datasets

provided a good starting point, none of the datasets is sufficient to encompass a wide range of realistic scenarios and to learn CNN based stronger pose estimation models. In this chapter, we build on these datasets and establish a large-scale benchmark with a much broader variety and an open evaluation setup. Our new dataset contains over $1.500\,00 \times 10^5$ annotated poses and over 2.2000×10^4 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 (Jhuang et al., 2013; Zhang et al., 2013; Charles et al., 2016) that focus on single isolated individuals 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 datasets that focus 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" (Sigal et al., 2010) and "Human3.6M" (Ionescu et al., 2014b) 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.

6.3. 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 our dataset from Chapter 5 and also the one introduced in (Insafutdinov et al., 2017). Similar to Chapter 5, 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. We use the same rationale to chose the videos as in Chapter 5, and selected video sequences 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.

6.3.1. Data Annotation

Since our goal in this chapter is to build a large-scale dataset, the selection of the annotation tool is crucial for efficient annotations. To scale-up the annotations efficiently, in this work, we chose the VATIC annotation tool (Vondrick et al., 2013). First, it allows to speed-up annotation by interpolating between frames. Second, VATIC is an online annotation tool which allowed us to employ a large number of annotators.

As before, we annotated the selected video sequences with person locations, identities and body pose. In Chapter 5, we carefully chose the videos such that they do not contain large crowds which are redundant to annotate. Further, we also annotated very tiny people by zooming them to a large

size. This, however, resulted in many cases where the annotations cannot be performed reliably due to extremely poor visibility. In contrast, in this chapter, we additionally annotated the ignore regions to exclude such crowds or people for which pose can not be reliably determined due to poor visibility. We performed the annotations in four steps. First, we labeled ignore regions. 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 the new 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. The poses for each person track are then annotated in the entire video. In contrast to Chapter 5, we annotate 15 body parts for each pose including head, nose, neck, shoulders, elbows, wrists, hips, knees and ankles. We added the nose to make the annotations compatible with the annotations provided by MSCOCO Keypoints dataset (Lin et al., 2014b). Further, we found that annotating the occlusion labels results in very high annotation time. Therefore, in this work, we chose to skip the annotation of the body joints that cannot be reliably localized by the annotator due to strong occlusion or difficult imaging conditions. This has proven to be a faster alternative to requiring annotators to guess the location of the joint and/or marking it as occluded. Figure 6.1 shows example frames from the dataset[2]. Note the variability in appearance and scale, and complexity due to a substantial number of people in close proximity.

Overall, the extended dataset contains 550 video sequences with 6.6374×10^4 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 this new 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. In contrast to Chapter 5, we did not annotate the entire sequences, but instead annotated 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. Further, this strategy allowed us to significantly reduce the annotation effort while also including more diverse scenarios.

In total, the extended dataset provides around 2.3000×10^4 labeled frames with $1.536\,15 \times 10^5$ 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 6.2 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 the plot of the track length in the middle-left is due to fixed length of the sequences included in the dataset.

[2]The example videos can be seen at https://www.youtube.com/watch?v=uYFRxGyMDe4

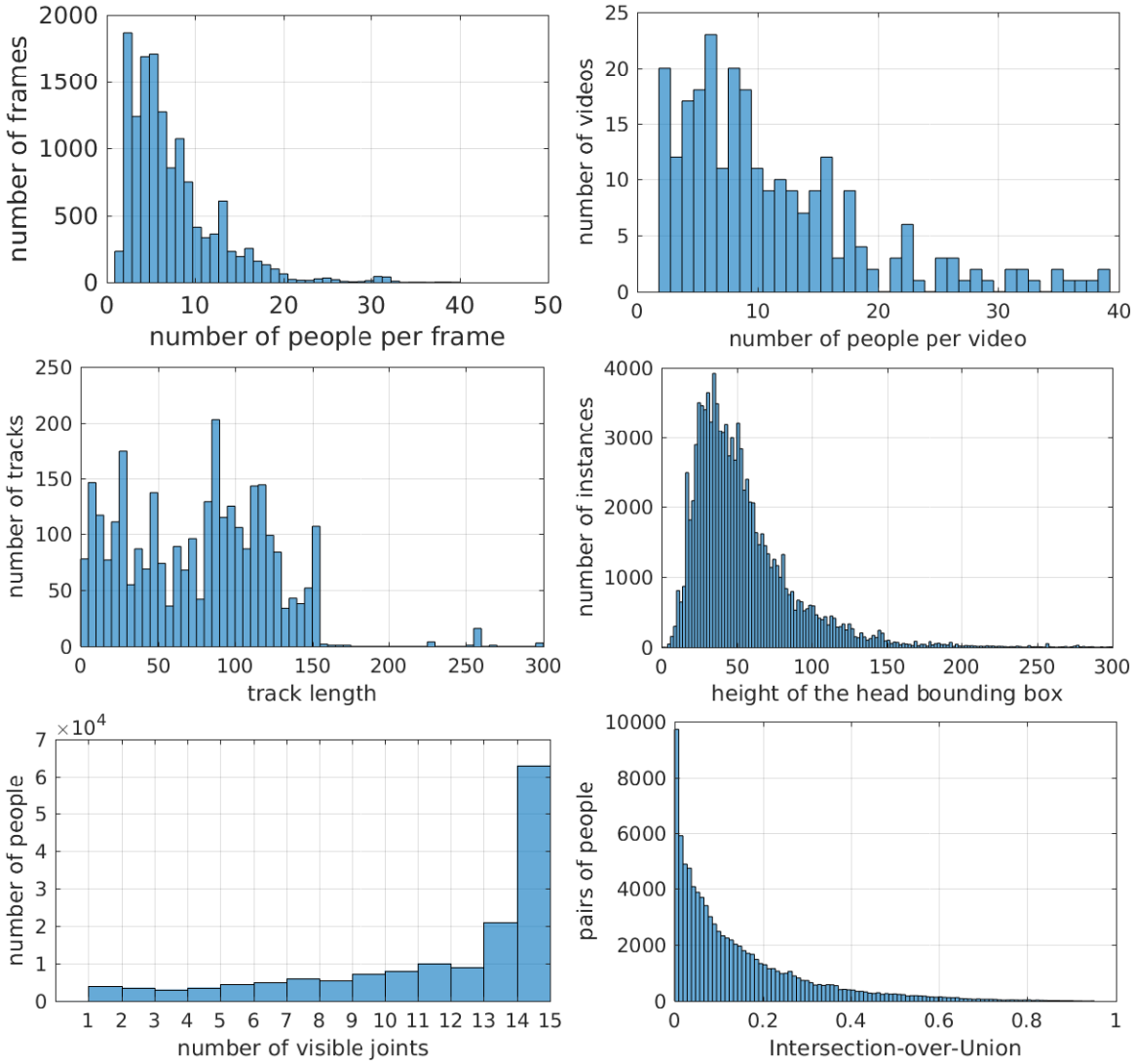


Figure 6.2: Various statistics of the PoseTrack benchmark.

6.3.2. 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 CLEAR MOT metrics (Bernardin and Stiefelhagen, 2008).

6.3.3. 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.

6.3.4. Experimental Setup and Evaluation Metrics

We follow the same evaluation protocol as proposed in Chapter 5. However, we made a few changes. First, in Chapter 5, we used a stricter version of PCKh and used 20% of the diagonal of the ground-truth head bounding box to consider a body joint to be correctly localized. However, in this chapter, we reverted back to the original threshold of 30% to make the evaluation compatible with MPII Pose Dataset (Andriluka et al., 2014). Second, since in this dataset, we did not annotate occlusion labels, we do not consider occlusion handling during the evaluation.

Same as in Chapter 5, we use mean Average Precision (mAP) (Pishchulin et al., 2016) to measure frame-wise multi-person pose accuracy. 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 averaging the per-joint metrics. 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, we will use MOTA as our main tracking metric. We decided to drop the trajectory-based measures (Li et al., 2009) used in Chapter 5 to ease the understanding of the performance of developed approaches.

The source code for the evaluation metrics is publicly available on the benchmark website.

6.4. ANALYSIS OF THE STATE-OF-THE-ART

As discussed earlier, multi-person pose estimation and tracking in unconstrained videos is a relatively new topic in computer vision research and only few approaches for this task have been proposed in the literature including our approach from Chapter 5 and (Insafutdinov et al., 2017). Therefore, to analyze the performance of the state-of-the-art on the extended dataset, we proceed in two ways.

First, we propose two baseline methods based on the approach presented in Chapter 5 and the approach in (Insafutdinov et al., 2017). Note that the extended benchmark includes an order of magnitude more sequences compared to the datasets used in Chapter 5 and (Insafutdinov et al., 2017), and the sequences in the new benchmark are about five times longer, which makes it computationally expensive to run the graph partitioning on the full sequences as done in these methods. We modify these methods to make them applicable on the longer sequences. The baselines and corresponding modifications are explained in Section 6.4.1.

Second, in order to broaden the scope of the evaluation we organized a PoseTrack Challenge in conjunction with ICCV'17 on the dataset by establishing an online evaluation server and inviting submissions from the research community. In the following we consider the top five methods submitted to the PoseTrack challenge for the pose estimation and pose tracking tasks. To make the

Submission	Pose model	Tracking model	Tracking granularity	Additional training data	mAP	MOTA
FlowTrack (Xiao et al., 2018)	modification of ResNet-152 (He et al., 2016)	Hungarian	pose-level	COCO	**74.6**	**57.8**
ProTracker (Girdhar et al., 2017)	Mask R-CNN (He et al., 2017)	Hungarian	pose-level	COCO	59.6	51.8
BUTD (Jin et al., 2017)	PAF (Cao et al., 2017)	graph partitioning	person-level and part-level	COCO	59.2	50.6
SOPT-PT (Zhong et al., 2017)	PAF (Cao et al., 2017)	Hungarian	pose-level	MPII-Pose + COCO	62.5	44.6
ML-LAB (Zhu et al., 2017)	modification of PAF (Cao et al., 2017)	frame-to-frame assign.	pose-level	MPII-Pose + COCO	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., 2017) + DeeperCut (Insafutdinov et al., 2016)	graph partitioning	pose-level	MPII-Pose + COCO	59.4	48.1
PoseTrack-baseline	PAF (Cao et al., 2017)	graph partitioning	part-level	COCO	59.4	48.4

Table 6.2: Results of the top five pose tracking models submitted to our evaluation server and of our baselines based on (Insafutdinov et al., 2017) and Chapter 5. 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
FlowTrack (Xiao et al., 2018)	modification of ResNet152 (He et al., 2016)	COCO	**77.15**
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 (Zhong et al., 2017)	PAF (Cao et al., 2017)	MPII-Pose + COCO	62.5
SSDHG	SSD (Liu et al., 2016) + 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 6.3: Results of the top five pose estimation models submitted to our evaluation server 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. 6.2 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 6.4: Pose estimation performance (mAP) of our ArtTrack baseline for different training sets.

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 6.5: Pose tracking performance (MOTA) of ArtTrack baseline for different part detection cut-off thresholds τ.

analysis more up-to-date, we also consider the current state-of-the-art approach of Xiao et al. (2018). In Table 6.2 and 6.3 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 Chapter 5 and (Insafutdinov et al., 2017), and then summarize the main observations made in this evaluation.

6.4.1. Baseline Methods

We build the first baseline model following the graph partitioning formulation for articulated tracking proposed in (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 (Insafutdinov et al., 2017) and Chapter 5. We use a publicly available Faster-RCNN (Ren et al., 2015) detector from the TensorFlow Object Detection API (Huang et al., 2017) 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 (Insafutdinov et al., 2016) 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 in (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 from (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 in (Insafutdinov et al., 2017) with additional features adding minimal improvements at the cost of substantially slower inference.

For the second baseline, we use our approach from Chapter 5 and replace the pose estimation model with (Cao et al., 2017). We empirically found that the pose estimation model of (Cao et al., 2017) is better at handling large scale variations compared to DeeperCut (Insafutdinov et al., 2016) used in the original approach. We do not make any changes in the graph partitioning algorithm, but reduce the window size to 21 as compared to 31 used in Chapter 5. The goal of constructing these strong baselines is to validate the results submitted to the evaluation server and to allow us to perform additional experiments presented in Section 6.5. In the rest of this chapter, we refer to them as ArtTrack-baseline and PoseTrack-baseline, respectively.

6.4.2. 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 (ResNet (He et al., 2016), Mask-RCNN (He et al., 2017), PAF (Cao et al., 2017)). On the other hand, when evaluating methods according to pose estimation metric (see Table 6.3) three of the top five approaches build on PAF (Cao et al., 2017), while the best performing approach (Xiao et al., 2018) uses a modification of ResNet-152 (He et al., 2016). The performance still varies considerably among the PAF-based methods (70.3 for submission ML-LAB (Zhu et al., 2017) vs. 62.5 for submission SOPT-PT (Zhong et al., 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 6.2. Only the submission 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. As shown in Chapter 4, this is suboptimal since multiple people overlap significantly, yet most approaches choose to ignore such cases (possibly for inference speed/efficiency reasons). The best performing approach FlowTrack (Xiao et al., 2018) relies on simple matching between frames based on Hungarian algorithm and matching cost based on OKS score between body poses propagated using optical-flow. While the winning entry of the PoseTrack Challenge, ProTracker (Girdhar et al., 2017), relies on 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. Note that the pose tracking performance of the top performing approach FlowTrack (Xiao et al., 2018) is 57.8 MOTA, with the remaining reasonable approaches showing rather similar MOTA results (51.8 for ProTracker (Girdhar et al., 2017) vs. 50.6 for BUTD (Jin et al., 2017) vs. 48.4 for PoseTrack-baseline (Chapter 5) vs. 48.1 for ArtTrack-baseline). The large gain in MOTA for the approach FlowTrack (Xiao et al., 2018) is likely due to a very strong pose estimation model (mAP 77.15 vs. 70.3 of entry ML-LAB) based on ResNet-152 (He et al., 2016).

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 followed by these methods is to pre-train on external data and then fine-tune on our training set. Our training set consists of 2.437×10^3 people tracks with 6.1178×10^4 annotated body poses and is complementary to COCO and MPII-Pose datasets which include an order of magnitude more individual people but do not provide motion information. We quantify the performance improvement due to training on additional data in Table 6.4 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 this 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 (Carreira and Zisserman, 2017a), inferring sequences of poses by propagating annotations from reliable keyframes (Charles et al., 2016), leveraging synthetic train-

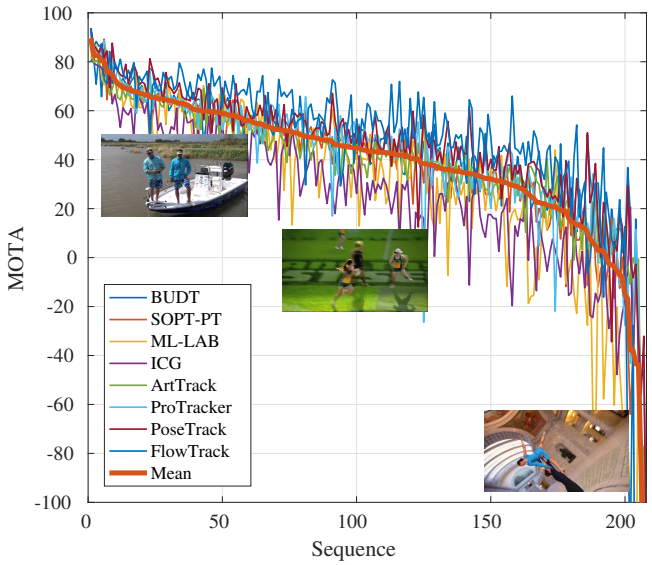


Figure 6.3: Sequences sorted by average MOTA.

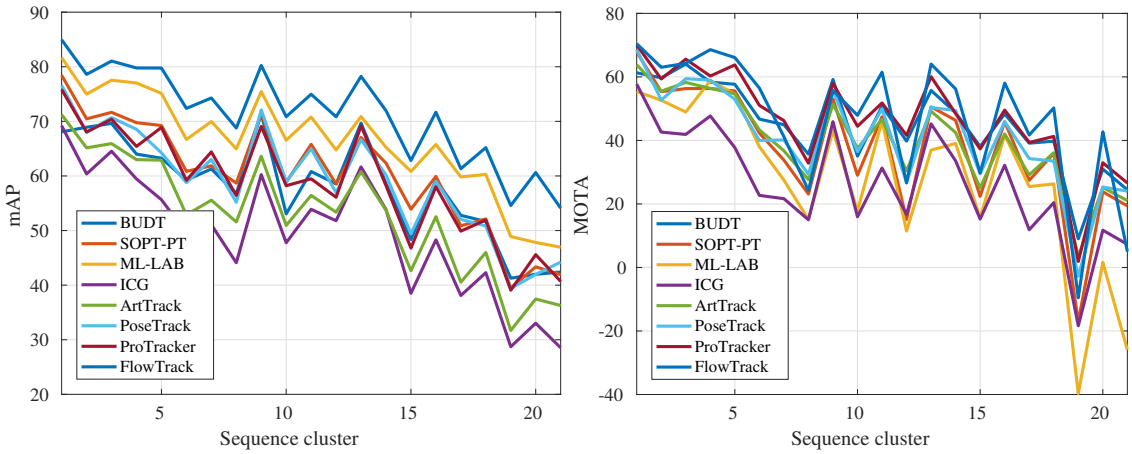


Figure 6.4: Pose estimation (left) and pose tracking (right) results sorted according to articulation complexity of the sequence.

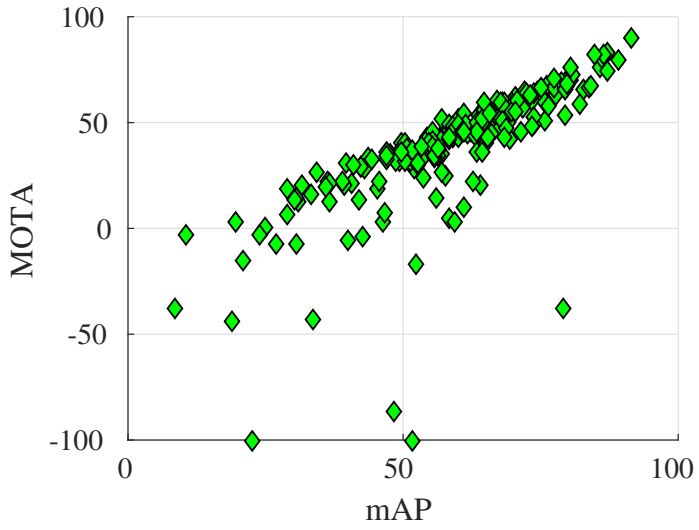


Figure 6.5: Visualization of correlation between mAP and MOTA for each sequence. Note the outliers in right plot that correspond to sequences where pose estimation works well but tracking still fails.

ing data as in (Varol et al., 2017), or learning correspondences between frames in an unsupervised way (Ranjan et al., 2018; Meister et al., 2017; Yin and Shi, 2018; Vondrick et al., 2018).

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 6.3 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. For example, for the best performing method FlowTrack (Xiao et al., 2018) the performance varies from nearly 90 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 likely to require methods that are beyond simple tracking component based on frame-to-frame assignment used in the currently best performing approaches.

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 (cf., performance of the submission ProTracker (Girdhar et al., 2017) in Table 6.2 and 6.3). We quantify this in Figure 6.5 for the 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 the 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

[3]Note that MOTA metric can become negative for example when the number of false positives significantly exceeds the number of ground-truth targets. (see Section. 3.3.4)

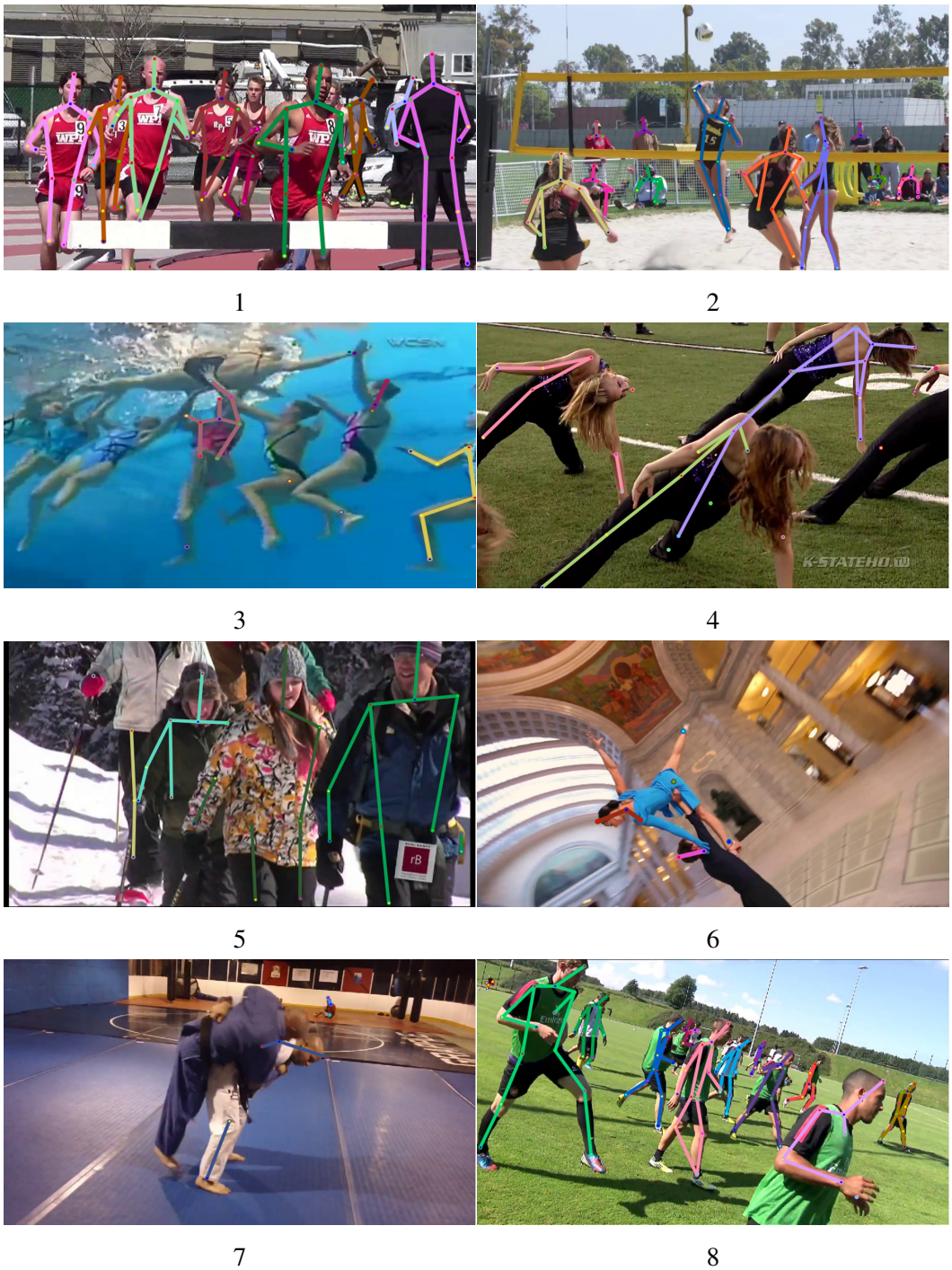
Figure 6.6: Selected frames from sample sequences with MOTA score above 75% with predictions of our ArtTrack-baseline overlaid in each frame. See text for further description.

the evaluation protocol to include confidence scores is an important future direction.

6.5. 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 eight 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 6.3 along with the original MOTA scores of each of the approaches. First, we observe that all methods perform similarly well on easy sequences. Figure 6.6 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 6.7 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 and MOTA is reported for each sequence. The result of this evaluation is shown in Figure 6.4. 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



1 2 3 4 5 6 7 8

Figure 6.7: 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).

with the increased pose complexity. Figure 6.5 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 6.7 (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. Additional examples and analyses of challenging sequences can be seen at https://www.youtube.com/watch?v=uYFRxGyMDe4.

6.6. SUMMARY

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.

CHAPTER 7

Pose for Action – Action for Pose

In Chapter 5 & 6 we discussed several methods for multi-person pose estimation and tracking. The methods provide pose trajectories for each person visible in the video. Such body pose trajectories have been shown to be useful for activity recognition in the literature (Jhuang et al., 2013; Chéron et al., 2015). Intuitively, the information about the activity of a person can also provide a strong cue about the pose. Hence, in this chapter we propose to utilize information about human actions to improve pose estimation.

We present a pictorial structure model that exploits high-level information about activities to incorporate higher-order part dependencies by modeling action specific appearance models and pose priors. However, instead of using an additional expensive action recognition framework, the action priors are efficiently estimated by our pose estimation framework, *i.e.*, by using the estimated pose trajectories. We also show that learning the right amount of appearance sharing among action classes improves the pose estimation. We demonstrate the effectiveness of the proposed method on two challenging datasets for pose estimation and action recognition with over 80,000 test images.

Contents

7.1. INTRODUCTION

Human pose estimation from RGB images or videos is a challenging problem in computer vision, especially for realistic and unconstrained data taken from the Internet. Popular approaches for pose estimation (Desai and Ramanan, 2012; Pishchulin et al., 2013b; Yang and Ramanan, 2013; Dantone

et al., 2014; Cherian et al., 2014; Tompson et al., 2014b) adopt the pictorial structure (PS) model, which resembles the human skeleton and allows for efficient inference in case of tree structures (Felzenszwalb and Huttenlocher, 2005; Felzenszwalb et al., 2010). Even if they are trained discriminatively, PS models struggle to cope with the large variation of human pose and appearance. This problem can be addressed by conditioning the PS model on additional observations from the image. For instance, (Pishchulin et al., 2013b) detects poselets, which are examples of consistent appearance and body part configurations, and condition the PS model on these.

Instead of conditioning the PS model on predicted configurations of body parts from an image, we propose to condition the PS model on high-level information like activity. Intuitively, the information about the activity of a person can provide a strong cue about the pose (Jhuang et al., 2013; Pishchulin et al., 2014; Chéron et al., 2015) and vice versa the activity can be estimated from pose. There are only few works (Yao et al., 2012b; Yu et al., 2013; Nie et al., 2015) that couple action recognition and pose estimation to improve pose estimation. In (Yao et al., 2012b), action class confidences are used to initialize an optimization scheme for estimating the parameters of a subject-specific 3D human model in indoor multi-view scenarios. In (Yu et al., 2013), a database of 3D poses is used to learn a cross-modality regression forest that predicts the 3D poses from a sequence of 2D poses, which are estimated using (Yang and Ramanan, 2013). In addition, the action is detected and the 3D poses corresponding to the predicted action are used to refine the pose. However, both approaches cannot be applied to unconstrained monocular videos. While Yao et al. (2012b) require a subject-specific model and several views, Yu et al. (2013) require 3D pose data for training. The closest to our work is the recent approach of Nie et al. (2015). They proposed an approach to jointly estimate action classes and refine human poses. The approach decomposes the human poses estimated at each video frame into sub-parts and tracks these sub-parts across time according to the parameters learned for each action. The action class and joint locations corresponding to the best part-tracks are selected as estimates for the action class and poses. The estimation of activities, however, comes at high computational cost since the videos are pre-processed by several approaches, one for pose estimation (Park and Ramanan, 2011) and two for extracting action related features (Wang et al., 2011, 2014b).

In this work, we present a framework for pose estimation that infers and integrates activities with a very small computational overhead compared to an approach that estimates the pose only. This is achieved by an action conditioned pictorial structure (ACPS) model for 2D human pose estimation that incorporates priors over activities. The framework of the approach is illustrated in Figure 7.1. We first infer the poses for each frame with a uniform distribution over actions. While the binaries of the ACPS are modeled by GMM, which depend on the prior distribution over the action classes, the unaries of the ACPS model are estimated by action conditioned regression forests. To this end, we modify the approach (Dantone et al., 2014), which consists of two layers of random forests, on two counts. Firstly, we replace the first layer by a convolutional network and use convolutional channel features to train the second layer, which consists of regression forests. Secondly, we condition the regression forests on a distribution over actions and learn the sharing among action classes. In our experiments, we show that these modifications increase the pose estimation accuracy by more than 40% compared to (Dantone et al., 2014). After the poses are inferred with a uniform distribution over actions, we update the action prior and the ACPS model based on the inferred poses to obtain the final pose estimates. Since the update procedure is very efficient, we avoid the computational expensive overhead of (Nie et al., 2015).

We evaluate our approach on the challenging J-HMDB (Jhuang et al., 2013) and Penn-

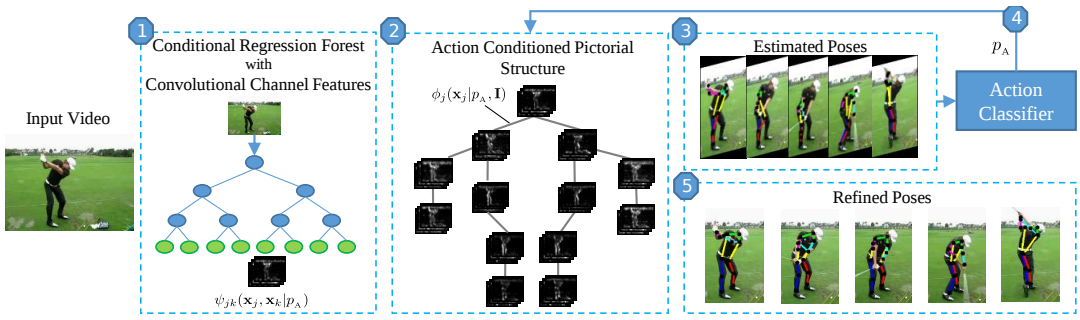


Figure 7.1: Overview of the proposed framework. We propose an action conditioned pictorial structure model for human pose estimation (2). Both the unaries ϕ and the binaries ψ of the model are conditioned on the distribution of action classes p_A. While the pairwise terms are modeled by Gaussians conditioned on p_A, the unaries are learned by a regression forest conditioned on p_A (1). Given an input video, we do not have any prior knowledge about the action and use a uniform prior p_A. We then predict the pose for each frame independently (3). Based on the estimated poses, the probabilities of the action classes p_A are estimated for the entire video (4). Pose estimation is repeated with the updated action prior p_A to obtain better pose estimates (5).

Action (Zhang et al., 2013) datasets, which consist of videos collected from the Internet and contain large amount of scale and appearance variations. In our experiments, we provide a detailed analysis of the impact of conditioning unaries and binaries on a distribution over actions and the benefit of appearance sharing among action classes. We demonstrate the effectiveness of the proposed approach for pose estimation and action recognition on both datasets. Compared to (Nie et al., 2015), the pose estimation accuracy is improved by over 30%. The models and source code are publicly available.[1]

7.2. OVERVIEW

Our method exploits the fact that the information about the activity of a subject provides a cue about pose and appearance of the subject, and vice versa. In this chapter we utilize the high-level information about a person's activity to leverage the performance of pose estimation, where the activity information is obtained from previously inferred poses. To this end, we propose an action conditioned pictorial structure (PS) that incorporates action specific appearance and kinematic models. If we have only a uniform prior over the action classes, the model is a standard PS model, which we will briefly discuss in Section 7.3. Figure 7.1 depicts an overview of the proposed framework.

7.3. PICTORIAL STRUCTURE

We adopt the joint representation (Dantone et al., 2014) of the PS model (Felzenszwalb and Huttenlocher, 2005), where the vector $\mathbf{x}_j \in \mathbf{p}$ represents the 2D location of the j^{th} joint in image $\mathbf{I}$, and $\mathbf{p} = \{\mathbf{x}_j\}_{j \in \mathcal{J}}$ is the set of all body joints. The structure of a human body is represented by a kinematic tree with nodes of the tree being the joints j and edges $\mathcal{E}$ being the kinematic constraints between a joint j and its unique parent joint k as illustrated in Figure 7.1. The pose configuration in a single

[1] http://pages.iai.uni-bonn.de/iqbal_umar/action4pose/

image is then inferred by maximizing the following posterior distribution:

$$p(\mathbf{p}|\mathbf{I}) \propto \prod_{j \in \mathcal{J}} \phi_j(\mathbf{x}_j|\mathbf{I}) \prod_{j,k \in \mathcal{E}} \psi_{jk}(\mathbf{x}_j, \mathbf{x}_k) \tag{7.1}$$

where the unary potentials $\phi_j(\mathbf{x}_j|\mathbf{I})$ represent the likelihood of the j^{th} joint at location $\mathbf{x}_j$. The binary potentials $\psi_{jk}(\mathbf{x}_j, \mathbf{x}_k)$ define the deformation cost for the joint-parent pair (j, k), and are often modeled by Gaussian distributions for an exact and efficient solution using a distance transform (Felzenszwalb and Huttenlocher, 2005). We describe the unary and binary terms in Section 7.3.1 and Section 7.3.2, respectively. In Section 7.4.1, we then discuss how these can be adapted to build an action conditioned PS model.

7.3.1. Unary Potentials

Random regression forests have been proven to be robust for the task of human pose estimation (Shotton et al., 2011b; Sun et al., 2012; Dantone et al., 2014). A regression forest $\mathcal{T}$ consists of a set of randomized regression trees, where each tree T is composed of split and leaf nodes. Each split node represents a weak classifier which passes an input image patch R to a subsequent left or right node until a leaf node L_T is reached. As in (Dantone et al., 2014), we use a separate regression forest for each body joint. Each tree is trained with a set of randomly sampled images from the training data. The patches around the annotated joint locations are considered as foreground and all others as background. Each patch consists of a joint label $c \in \mathcal{J}$, a set of image features $\mathbf{g}_R$, and its 2D offset $\mathbf{d}_R$ from the joint center. During training, a splitting function is learned for each split node by randomly selecting and maximizing a goodness measure for regression or classification. At the leaves the class probabilities $p(c|L_T)$ and the distribution of offset vectors $p(\mathbf{d}|L_T)$ are stored.

During testing, patches are densely extracted from the input image $\mathbf{I}$ and are passed through the trained trees. Each patch centered at location $\mathbf{y}$ ends in a leaf node $L_T(R(\mathbf{y}))$ for each tree $T \in \mathcal{T}$. The unary potentials ϕ_j for the joint j at location $\mathbf{x}_j$ are then given by

$$\phi_j(\mathbf{x}_j|\mathbf{I}) = \sum_{\mathbf{y} \in \mathcal{X}} \frac{1}{|\mathcal{T}|} \sum_{T \in \mathcal{T}} \Big\{ p(c{=}j|L_T(R(\mathbf{y}))) \cdot p(\mathbf{x}{-}\mathbf{y}|L_T(R(\mathbf{y})) \Big\}. \tag{7.2}$$

In (Dantone et al., 2014) a two layer approach is proposed. The first layer consists of classification forests that classify image patches according to the body parts using a combination of color features, HOG features, and the output of a skin color detector. The second layer consists of regression forests that predict the joint locations using the features of the first layer and the output of the first layer as features. For both layers, the split nodes compare feature values at different pixel locations within a patch of size 24×24 pixels.

We propose to replace the first layer by a convolutional network and extract convolutional channel features (CCF) (Yang et al., 2015) from the intermediate layers of the network to train the regression forests of the second layer. In (Yang et al., 2015) several pre-trained network architectures have been evaluated for pedestrian detection using boosting as classifier. The study shows that the "conv3-3" layer of the VGG-16 net (Simonyan and Zisserman, 2014) trained on the ImageNet (ILSVRC-2012) dataset performs very well even without fine tuning, but it is indicated that the optimal layer depends on the task. Instead of pre-selecting a layer, we use regression forests to select the features based on the layers "conv2-2", "conv3-3", "conv4-3", and "conv5-3". An example of the CCF extracted

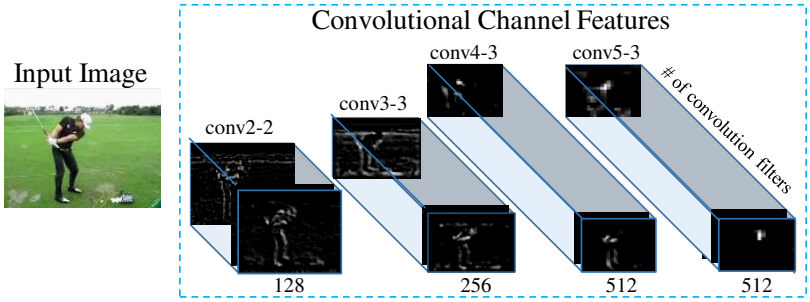


Figure 7.2: Example of convolutional channel features extracted using VGG-16 net (Simonyan and Zisserman, 2014).

from an image is shown in Figure 7.2. Since these layers are of lower dimensions than the original image, we upsample them using linear interpolation to make their dimensions equivalent to the input image. This results in a 1408 (128+256+512+512) dimensional feature representation for each pixel. As split nodes in the regression forests, we use axis-aligned split functions. For an efficient feature extraction at multiple image scales, we use patchwork as proposed in (Iandola et al., 2014) to perform the forward pass of the convolutional network only once.

7.3.2. Binary Potentials

Binary potentials $\psi_{jk}(\mathbf{x}_j, \mathbf{x}_k)$ are modeled as a Gaussian mixture model for each joint j with respect to its parent joint k in the kinematic tree. As in (Dantone et al., 2014), we obtain the relative offsets between child and parent joints from the training data and cluster them into $m = 1, \ldots, M$ clusters using k-means clustering. Each cluster m takes the form of a weighted Gaussian distribution as

$$\psi_{jk}(\mathbf{x}_j, \mathbf{x}_k) = \beta_{jk}^m \cdot \exp\left(-\frac{1}{2}\left(\mathbf{d}_{jk} - \mu_{jk}^m\right)^T \left(\Sigma_{jk}^m\right)^{-1} \left(\mathbf{d}_{jk} - \mu_{jk}^m\right)\right) \tag{7.3}$$

with mean μ_{jk}^m and covariance Σ_{jk}^m, where $\mathbf{d}_{jk} = (\mathbf{x}_j - \mathbf{x}_k)$. The weights β_{jk}^m are set according to the cluster frequency $p(m|j,k)^\alpha$ with a normalization constant $\alpha = 0.1$ (Dantone et al., 2014).

For inference, we select the best cluster m for each joint by computing the max-marginals for the root node and backtrack the best pose configuration from the maximum of the max-marginals.

7.4. ACTION CONDITIONED POSE ESTIMATION

As illustrated in Figure 7.1, our goal is to estimate the pose $\mathbf{p}$ conditioned by the distribution p_{A} for a set of action classes $a \in \mathcal{A}$. To this end, we introduce in Section 7.4.1 a pictorial structure model that is conditioned on p_{A}. Since we do not assume any prior knowledge of the action, we estimate the pose first with the uniform distribution $p_{\text{A}}(a) = 1/|\mathcal{A}|, \forall a \in \mathcal{A}$. The estimated poses for F frames are then used to estimate the probabilities of the action classes $p_{\text{A}}(a|\mathbf{p}_{f=1\ldots F}), \forall a \in \mathcal{A}$ as described in Section 7.4.2. Finally, the poses $\mathbf{p}_f$ are updated based on the distribution p_{A}.

Figure 7.3: Example patches centered at the wrist of the left hand side. We can see a large amount of appearance variation for a single body part. However, for several activities, in particular sports such as *golf* and *pull-up*, this variation is relatively small within the action classes. Nonetheless, a few classes also share appearance with each other e.g., *golf* and *baseball* or activities such as *run* and *kick ball*. This clearly shows the importance of class specific appearance models with a right amount of appearance sharing across action classes for efficient human pose estimation.

7.4.1. Action Conditioned Pictorial Structure

In order to integrate the distribution p_{A} of the action classes obtained from the action classifier into (7.1), we make the unaries and binaries dependent on p_{A}:

$$p(\mathbf{p}|p_{\text{A}},\mathbf{I}) \propto \prod_{j\in\mathcal{J}} \phi_j(\mathbf{x}_j|p_{\text{A}},\mathbf{I}) \cdot \prod_{j,k\in\mathcal{E}} \psi_{jk}(\mathbf{x}_j,\mathbf{x}_k|p_{\text{A}}). \tag{7.4}$$

While the unary terms are discussed in Section 7.4.1.1, the binaries $\psi_{jk}(\mathbf{x}_j,\mathbf{x}_k|p_{\text{A}})$ are represented by Gaussians as in (7.3). However, instead of computing mean and covariance from all training poses with equal weights, we weight each training pose based on its action class label and $p_{\text{A}}(a)$. In our experiments, we will also investigate the case where $p_{\text{A}}(a)$ is simplified to

$$p_{\text{A}}(a) = \begin{cases} 1 & \text{if} \quad a = \arg\max_{a'} p_{\text{A}}(a'|\mathbf{p}_{f=1\ldots F}) \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

7.4.1.1. Conditional Joint Regressors

Figure 7.3 shows examples of patches of the wrist extracted from images of different action classes. We can see a large amount of appearance variation across different classes regardless of the fact that all patches belong to the same body joint. However, it can also be seen that within individual activities this variation is relatively small. We exploit this observation and propose action specific unary potentials for each joint j. To this end we adopt conditional regression forests (Dantone et al., 2012; Sun et al., 2012) that have been proven to be robust for facial landmark detection in (Dantone et al., 2012) and 3D human pose estimation in (Sun et al., 2012). While (Dantone et al., 2012) trains a separate regression forest for each head pose and selects a specific regression forest conditioned

on the output of a face pose detector, (Sun et al., 2012) proposes partially conditioned regression forests, where a forest is jointly trained for a set of discrete states of a human attribute like human orientation or height and the conditioning only happens at the leaf nodes. Since the continuous attributes are discretized, interpolation between the discrete states is achieved by sharing the votes. In this we resort to partially conditional forests due to its significantly reduced training time and memory requirements. During training we augment each patch R with its action class label a. Instead of $p(c|L_T)$ and $p(\mathbf{d}|L_T)$, the leaf nodes model the conditional probabilities $p(c|a, L_T)$ and $p(\mathbf{d}|a, L_T)$. Given the distribution over action classes p_{A}, we obtain the conditional unary potentials:

$$\begin{aligned}\phi_j(\mathbf{x}_j|p_{\text{A}}, \mathbf{I}) &= \sum_{\mathbf{y}\in\mathcal{X}} \frac{1}{|\mathcal{T}|} \sum_{T\in\mathcal{T}} \sum_{a\in\mathcal{A}} \Big\{ p_{\text{A}}(a) \cdot p(c = j|a, L_T(R(\mathbf{y}))).p(\mathbf{x} - \mathbf{y}|a, L_T(R(\mathbf{y})) \Big\} \\ &= \sum_{a\in\mathcal{A}} \phi_j(\mathbf{x}_j|a, \mathbf{I}) p_{\text{A}}(a). \end{aligned} \tag{7.6}$$

Since the terms $\phi_j(\mathbf{x}_j|a, \mathbf{I})$ need to be computed only once for an image $\mathbf{I}$, $\phi_j(\mathbf{x}_j|p_{\text{A}}, \mathbf{I})$ can be efficiently computed after an update of p_{A}.

7.4.1.2. Appearance Sharing Across Actions

Different actions sometimes share body pose configurations and appearance of parts as shown in Figure 7.3. We therefore propose to learn the sharing among action classes within a conditional regression forest. To this end, we replace the term $\phi_j(\mathbf{x}_j|a, \mathbf{I})$ in (7.6) by a weighted combination of the action classes:

$$\phi_j^{sharing}(\mathbf{x}_j|a, \mathbf{I}) = \sum_{a'\in\mathcal{A}} \gamma_a(a') \phi_j(\mathbf{x}_j|a', \mathbf{I}) \tag{7.7}$$

where the weights $\gamma_a(a')$ represent the amount of sharing between action class a and a'. To learn the weights γ_a for each class $a \in \mathcal{A}$, we apply the trained conditional regression forests to a set of validation images scaled to a constant body size and maximize the response of (7.7) at the true joint locations and minimize it at non-joint locations. Formally, this can be stated as

$$\begin{aligned}\gamma_a = \arg\max_{\gamma} \sum_{n_a} \sum_{j} \Bigg\{ &\sum_{a'\in\mathcal{A}} \gamma(a') \phi_j^* \left(\mathbf{x}_{j,n_a}^{gt} | a', \mathbf{I}_{n_a} \right) \\ &- \max_{\mathbf{x}\in\mathbf{X}_{j,n_a}^{neg}} \left(\sum_{a'\in\mathcal{A}} \gamma(a') \phi_j^* \left(\mathbf{x}|a', \mathbf{I}_{n_a} \right) \right) \Bigg\} - \lambda \|\gamma\|^2 \end{aligned} \tag{7.8}$$

subject to $\sum_{a'\in\mathcal{A}} \gamma(a') = 1$ and $\gamma(a') \geqslant 0$. $\mathbf{I}_{n_a}$ denotes the n^{th} scaled validation image of action class a, $\mathbf{x}_{j,n_a}^{gt}$ is the annotated joint position for joint j in image $\mathbf{I}_{n_a}$, and $\mathbf{X}_{j,n_a}^{neg}$ is a set of image locations which are more than 5 pixels away from $\mathbf{x}_{j,n_a}^{gt}$. The set of negative samples is obtained by computing $\phi_j^*(\mathbf{x}|a', \mathbf{I}_{n_a})$ and taking the 10 strongest modes, which do not correspond to $\mathbf{x}_{j,n_a}^{gt}$, for each image. For optimization, we use the smoothed unaries

$$\phi_j^*(\mathbf{x}|a, \mathbf{I}) = \sum_{\mathbf{y}\in\mathcal{X}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{\sigma^2} \right) \phi_j(\mathbf{y}|a, \mathbf{I}) \tag{7.9}$$

with $\sigma = 3$ and replace $\max$ by the softmax function to make (7.8) differentiable. The last term in (7.8) is a regularizer that prefers sharing, *i.e.*, $\|\gamma\|^2$ attains its minimum value for uniform weights. In our experiments, we use $\lambda = 0.4$ as weight for the regularizer. We optimize the objective function by constrained local optimization using uniform weights for initialization $\gamma(a') = 1/|\mathcal{A}|$. In our experiments, we observed that similar weights are obtained when the optimization is initialized by $\gamma(a) = 1$ and $\gamma(a') = 0$ for $a' \neq a$, indicating that the results are not sensitive to the initialization. In (7.8), we learn the weights γ_a for each action class but we could also optimize for each joint independently. In our experiments, however, we observed that this resulted in over-fitting.

7.4.2. Action Classification

For pose-based action recognition, we use the bag-of-word approach proposed in (Jhuang et al., 2013). From the estimated joint positions $\mathbf{p}_{f=1\ldots F}$, a set of features called *NTraj+* is computed that encodes spatial and directional joint information. Additionally, differences between successive frames are computed to encode the dynamics of the joint movements. Since we use a body model with 13 joints, we compute the locations of missing joints (neck and belly) in order to obtain the same 15 joints as in (Jhuang et al., 2013). We approximate the neck position as the mean of the face and the center of shoulder joints. The belly position is approximated by the mean of the shoulder and hip joints.

For each of the $3,223$ descriptor types, a codebook is generated by running k-means 8 times on all training samples and choosing the codebook with minimum compactness. These codebooks are used to extract a histogram for each descriptor type and video. For classification, we use an SVM classifier in a multi-channel setup. To this end, for each descriptor type t, we compute a distance matrix D_t that contains the χ^2-distance between the histograms (h_i^t, h_j^t) of all video pairs $(\mathcal{F}_i, \mathcal{F}_j)$. We then obtain the kernel matrix that we use for the multi-class classification as follows

$$K(\mathcal{F}_i, \mathcal{F}_j) = \exp\left(-\frac{1}{L}\sum_{t=1}^{L}\frac{D_t(h_i^t, h_j^t)}{\mu^t}\right) \tag{7.10}$$

where L is the number of descriptor types and μ^t is the mean of the distance matrix D_t. For classification we use a one-vs-all approach with $C = 100$ for the SVM.

7.5. EXPERIMENTS

In order to evaluate the proposed method, we follow the same protocol as proposed in (Nie et al., 2015). In particular, we evaluate the proposed method on two challenging datasets, namely sub-J-HMDB (Jhuang et al., 2013) and the Penn-Action dataset (Zhang et al., 2013). Both datasets provide annotated 2D poses and activity labels for each video. They consist of videos collected from the Internet and contain large amount of scale and appearance variations, low resolution frames, occlusions and foreshortened body poses. This makes them very challenging for human pose estimation. While sub-J-HMDB (Jhuang et al., 2013) comprises videos from 12 action categories with fully visible persons, the Penn-Action dataset comprises videos from 15 action categories with a large amount of body part truncations. As in (Nie et al., 2015), we discard the activity class “playing guitar" since it does not contain any fully visible person. For testing on sub-J-HMDB, we follow the 3-fold cross validation protocol proposed by (Jhuang et al., 2013). On average for each split, this

includes 229 videos for training and 87 videos for testing with 8,124 and 3,076 frames, respectively. The Penn-Action dataset consists of 1,212 videos for training and 1,020 for testing with 85,325 and 74,308 frames, respectively. To evaluate the performance of pose estimation, we use the PCK (Percentage of Correct Keypoints) metric (Yang and Ramanan, 2013; Nie et al., 2015).

Features	
HOG, Color, Skin (Dantone et al., 2014)	CCF
36.7	51.5

Table 7.1: Comparison of the features used in (Dantone et al., 2014) with the proposed convolutional channel features (CCF). PCK with threshold 0.1 on split-1 of sub-J-HMDB.

Unary \ **Binary**	Indep.	Cond. (7.5)	Cond. (p_{A})
Indep. + CCF	51.5	53.8	51.0
Cond. (7.5) + CCF	48.9	49.9	48.4
Cond. (7.5) + AS + CCF	53.8	**55.3**	52.9
Cond. (p_{A}) + CCF	52.3	53.1	52.0
Cond. (p_{A}) + AS + CCF	53.4	**55.1**	52.5

(a)

Unary \ **Binary**	Indep.	Cond. (7.5)	Cond. (p_{A})
Indep.	36.7	38.5	36.7
Cond. (7.5)	29.3	32.5	29.7
Cond. (7.5) + AS	38.0	**39.6**	37.2
Cond. (p_{A})	37.0	39.0	36.8
Cond. (p_{A}) + AS	38.0	**39.5**	37.3

(b)

Table 7.2: Analysis of the proposed framework under different settings. Cond. (5) denotes if the action class probabilities p_{A} are replaced by (7.5). **(a)** using CCF features. **(b)** using features from (Dantone et al., 2014). (PCK: threshold: 0.1)

7.5.1. Implementation Details

For the Penn-Action dataset, we split the training images half and half into a training set and a validation set. Since the dataset sub-J-HMDB is smaller, we create a validation set by mirroring the training images. The training images are scaled such that the mean upper body size is 40 pixels. Each forest consists of 20 trees, where 10 trees are trained on the training and 10 on the validation set, with a maximum depth of 20 and a minimum of 20 patches per leaf. We train each tree with 50,000 positive and 50,000 negative patches extracted from 5,000 randomly selected images and generate 40,000 split functions at each node. For the binary potentials, we use $k = 24$ mixtures per part.

For learning the appearance sharing among action classes (Section 7.4.1.2) and training the action classifier (Section 7.4.2), we use the 10 trees trained on the training set and apply them to the validation set. The action classifier and the sharing are then learned on the validation set.

For pose estimation, we create an image pyramid and perform inference at each scale independently. We then select the final pose from the scale with the highest posterior (7.4). In our experiments, we use 4 scales with scale factor 0.8. The evaluation of 260 trees (20 trees for each of the 13 joints) including feature extraction takes roughly 15 seconds on average.[2] Inference with the PS model for all 4 scales takes around 1 second. The action recognition with feature computation takes only 0.18 seconds per image and it does not increase the time for pose estimation substantially.

Method	**Head**	**Sho**	**Elb**	**Wri**	**Hip**	**Knee**	**Ank**	**Avg** thr.=0.2	**Avg** thr.=0.1
Cond(7.5)+AS U. & Cond(7.5) B.+CCF	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8	51.6
Cond. (p_A)+AS U. & Cond(7.5) B.+CCF	90.1	76.7	59.2	54.7	85.6	76.2	72.9	73.6	51.2
Indep. U. & Indep. B.+CCF	88.1	76.3	57.0	49.2	85.0	75.4	71.7	71.8	48.7
State-of-the-art approaches									
Dantone et al. (2014)	65.6	56.4	39.1	31.1	65.2	62.8	60.9	54.4	34.4
Yang and Ramanan (2013)	73.8	57.5	30.7	22.1	69.9	58.2	48.9	51.6	—
Park and Ramanan (2011)	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5	—
Cherian et al. (2014)	47.4	18.2	0.08	0.07	—	—	—	16.4	—
Nie et al. (2015)	80.3	63.5	32.5	21.6	76.3	62.7	53.1	55.7	—
Chen and Yuille (2014)	78.7	68.4	48.3	39.7	76.3	66.3	60.3	62.6	42.2
Wei et al. (2016)	98.4	94.7	85.5	81.7	97.9	94.9	90.3	91.9	—
Song et al. (2017)	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1	—
Luo et al. (2018)	98.2	96.5	89.6	86.0	98.7	95.6	90.9	93.6	—

Table 7.3: Comparison with the state-of-the-art on sub-J-HMDB using PCK threshold 0.2. In the last column, the average accuracy for the threshold 0.1 is given.

7.5.2. Pose Estimation

We first evaluate the impact of the convolutional channel features (CCF) for pose estimation on split-1 of sub-J-HMDB. The results in Table 7.1 show that the CCF outperform the combination of color features, HOG features, and the output of a skin color detector, which is used in (Dantone et al., 2014).

In Table 7.2a we evaluate the proposed ACPS model under different settings on split-1 of sub-J-HMDB when using CCF features for joint regressors. We start with the first step of our framework where neither the unaries nor the binaries depend on the action classes. This is equivalent to the standard PS model described in Section 7.3, which achieves an average joint estimation accuracy of 51.5%. Given the estimated poses, the pose-based action recognition approach described in Section 7.4.2 achieves an action recognition accuracy of 66.3% for split-1.

Having estimated the action priors p_A, we first evaluate action conditioned binary potentials while keeping the unary potentials as in the standard PS model. As described in Section 7.4.1, we can use in our model the probabilities p_A or replace them by the distribution (7.5), which considers only the classified action class. The first setting is denoted by "Cond. (p_A)" and the second by "Cond.

[2] Measured on a 3.4GHz Intel processor using only one core with NVidia GeForce GTX 780 GPU. The image size for all videos in sub-J-HMDB is 320 × 240 pixels.

Method	Head	Sho	Elb	Wri	Hip	Knee	Ank	Avg thr.=0.2	Avg thr.=0.1
Cond(7.5)+AS U. & Cond(7.5) B.+CCF	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1	64.8
Indep. U. & Indep. B.+CCF	84.5	81.3	66.2	62.6	82.4	75.1	76.5	75.5	57.3
State-of-the-art approaches									
Yang and Ramanan (2013)	57.9	51.3	30.1	21.4	52.6	49.7	46.2	44.2	—
Park and Ramanan (2011)	62.8	52.0	32.3	23.3	53.3	50.2	43.0	45.3	—
Nie et al. (2015)	64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0	—
Gkioxari et al. (2016)	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8	—
Song et al. (2017)	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.5	—
Wei et al. (2016)	98.6	97.9	95.9	95.8	98.1	97.3	96.6	97.1	—
Luo et al. (2018)	98.9	98.6	96.6	96.6	98.2	98.2	97.5	97.7	—

Table 7.4: Comparison with the state-of-the-art in terms of joint localization error on the Penn-Action dataset.

(7.5)". It can be seen that the conditional binaries based on (7.5) already outperform the baseline by improving the accuracy from 51.5% to 53.8%. However, taking the priors from all classes slightly decreases the performance. This shows that conditioning the binary potentials on the most probable class is a better choice than using priors from all classes.

Secondly, we analyze how action conditioned unary potentials affect pose estimation. For the unaries, we have the same options "Cond. (p_{A})" and "Cond. (7.5)" as for the binaries. In addition, we can use appearance sharing as described in Section 7.4.1.2, which is denoted by "AS". For all three binaries, the conditional unaries based on (7.5) decrease the performance. Since the conditional unaries based on (7.5) are specifically designed for each action class, they do not generalize well in case of a misclassified action class.. However, adding appearance sharing to the conditional unaries boost the performance for both conditioned on (7.5) and p_{A}. Adding appearance sharing outperforms all other unaries without appearance sharing, *i.e.*, conditional unaries, independent unaries and the unaries conditioned on p_{A}. For all unaries, the binaries conditioned on (7.5) outperform the other binaries. This shows that appearance sharing and binaries conditioned on the most probable class performs best, which gives an improvement of the baseline from 51.5% to 55.3%.

In Table 7.2b, we also evaluate the proposed ACPS when using the weaker features from (Dantone et al., 2014). Although the accuracies as compared to CCF features are lower, the benefit of the proposed method remains the same. For the rest of this chapter, we will use CCF for all our experiments.

In Table 7.3 we compare the proposed action conditioned PS model with other state-of-the-art approaches on all three splits of sub-J-HMDB. In particular, we provide a comparison with (Dantone et al., 2014; Yang and Ramanan, 2013; Nie et al., 2015; Park and Ramanan, 2011; Cherian et al., 2014; Chen and Yuille, 2014). The accuracies for the approaches (Yang and Ramanan, 2013; Nie et al., 2015; Park and Ramanan, 2011; Cherian et al., 2014) are taken from (Nie et al., 2015) where the PCK threshold 0.2 is used. We also evaluated the convolutional network based approach (Chen and Yuille, 2014) using the publicly available source code trained on sub-J-HMDB. Our approach outperfroms the other methods by a margin, and notably improves wrist localization by more than 5% as compared to the baseline. Some qualitative results along with some failure cases on sub-JHMDB dataset can be seen in 7.4 and 7.6, respectively.

Table 7.4 compares the proposed ACPS with the state-of-the-art on the Penn-Action dataset. The

	Indep. U. + Indep. B. + CCF	Cond. (7.5)+AS U. & Cond. (7.5) B. + CCF			
		Pose	IDT-FV	Pose+IDT-FV	GT
sub-J-HMDB (split-1)	51.5	55.3 (56.2%)	52.6 (66.3%)	55.3 (76.4%)	55.9 (100%)
Penn-Action	57.3	64.8 (79.0%)	—	—	68.1 (100%)

Table 7.5: Analysis of pose estimation accuracy with respect to action recognition accuracy. The values in the parentheses are the corresponding action recognition accuracies. (PCK threshold: 0.1)

accuracies for the approaches (Yang and Ramanan, 2013; Nie et al., 2015; Park and Ramanan, 2011) are taken from (Nie et al., 2015). We can see that the proposed method improves the baseline from 75.5% to 81.1%, while improving the elbow and wrist localization accuracy by more than 7% and 10%, respectively. The proposed method also significantly outperforms other approaches.

We also compare our approach with the more recent CNN based approaches (Gkioxari et al., 2016; Song et al., 2017; Wei et al., 2016; Luo et al., 2018) on both datasets. The accuracies for the approach of Wei et al. (2016) are taken from (Luo et al., 2018). These methods achieve significantly better accuracies than our method. This is mainly due to the better multi-staged CNN architectures used as baselines by these methods compared to our network for computing CCF features. Since the gain of ACPS compared to our baseline even increases when better features are used as shown in Table 7.2a & Table 7.2b, we expect at least a similar performance gain when we use similar baseline architecture as in (Gkioxari et al., 2016; Song et al., 2017; Luo et al., 2018) for ACPS. Some qualitative results along with some failure cases on Penn-Action dataset can be seen in 7.5 and 7.7, respectively.

7.5.3. Action Recognition

In Table 7.6, we compare the action recognition accuracy obtained by our approach with state-of-the-approaches for action recognition. On sub-J-HMDB, the obtained accuracy using only pose as feature is comparable to the other approaches. Only the recent work (Chéron et al., 2015) which combines pose, CNN, and motion features achieves a better action recognition accuracy. However, if we combine our pose-based action recognition with Fisher vector encoding of improved dense trajectories (Wang and Schmid, 2013) using late fusion, we outperform other methods that also combine pose and appearance. The results are similar on the Penn-Action dataset.

In Table 7.5, we report the effect of different action recognition approaches on pose estimation. We report the pose estimation accuracy for split-1 of sub-J-HMDB when the action classes are not inferred by our framework, but estimated using improved dense trajectories with Fisher vector encoding (IDT-FV) (Wang and Schmid, 2013) or the fusion of our pose-based method and IDT-FV. Although the action recognition rate is higher when pose and IDT-FV are combined, the pose estimation accuracy is not improved. If the action classes are not predicted but are provided (GT), the accuracy improves slightly for sub-J-HMDB and from 64.8% to 68.1% for the Penn-Action dataset. We also experimented with several iterations in our framework, but the improvements compared to the achieved accuracy of 51.6% were not more than 0.1% on all three splits of sub-J-HMDB.

Method	sub-J-HMDB	Penn-Action
Appearance features only		
Dense (Jhuang et al., 2013)	46.0%	—
IDT-FV (Wang and Schmid, 2013)	60.9%	92.0%
Pose features only		
Pose (Jhuang et al., 2013)	54.1%	—
Pose (Ours)	61.5%	79.0%
Pose + Appearance features		
MST (Wang et al., 2014b)	45.3%	74.0%
Pose+Dense (Jhuang et al., 2013)	52.9%	—
AOG (Nie et al., 2015)	61.2%	85.5%
P-CNN (Chéron et al., 2015)	66.8%	—
Pose (Ours)+IDT-FV	74.6%	92.9%

Table 7.6: Comparison of action recognition accuracy with the state-of-the-art approaches on sub-J-HMDB and Penn-Action datasets.

7.6. SUMMARY

In this chapter, we have demonstrated that action recognition can be efficiently utilized to improve human pose estimation on realistic data. To this end, we presented a pictorial structure model that incorporates high-level activity information by conditioning the unaries and binaries on a prior distribution over action labels. Although the action priors can be estimated by an accurate, but expensive action recognition system, we have shown that the action priors can also be efficiently estimated during pose estimation without substantially increasing the computational time of the pose estimation. In our experiments, we thoroughly analyzed various combinations of unaries and binaries and showed that learning the right amount of appearance sharing among action classes improves the pose estimation accuracy. We demonstrated the effectiveness of proposed method on two challenging datasets for pose estimation and action recognition.

While we have successfully validated the idea of using action recognition for pose estimation, in contrast to previous chapters, we chose the PS model as a baseline. There are two main reasons for this choice. First, the available datasets that provide both the pose annotations and activity labels are not sufficiently large to train very deep multi-staged networks. For example, sub-JHMDB provides only 229 training videos for 12 action classes, *i.e.*, 20 videos per action class. Further, in contrast to image datasets such as MPII Pose dataset (Andriluka et al., 2014), video datasets contain far less appearance variation to train the deep CNNs effectively. Even the state-of-the-art method (Luo et al., 2018) for pose estimation in videos uses a model pre-trained on MPII pose dataset. Whereas, the regression forests with a pre-trained VGG model for feature extraction can be trained with lesser training data without over-fitting. Second, the CNN based models are often trained in an end-to-end way and directly integrating action probabilities in these models is less straightforward. In contrast, the PS model with regression forest based joint detectors allows to integrate activity information in a straightforward way, *i.e.*, by storing the action labels for each training patch at the leaf nodes of the forest and conditioning the binary potentials on the most probable action class.

We expect that further improvements can be achieved by integrating action information in the CNN based models. This, however, means that we need a larger dataset with pose and activity annotations. Further, we also need an adequate representation of action priors that can be integrated easily in the commonly adopted fully-convolutional neural networks. As done in this chapter, this

representation should not increase the runtime or the complexity of the model with an increase in the number of activities. We leave this as an interesting future work.

Figure 7.4: Qualitative results on some frames of sub-J-HMDB as compared to our baseline with CCF.

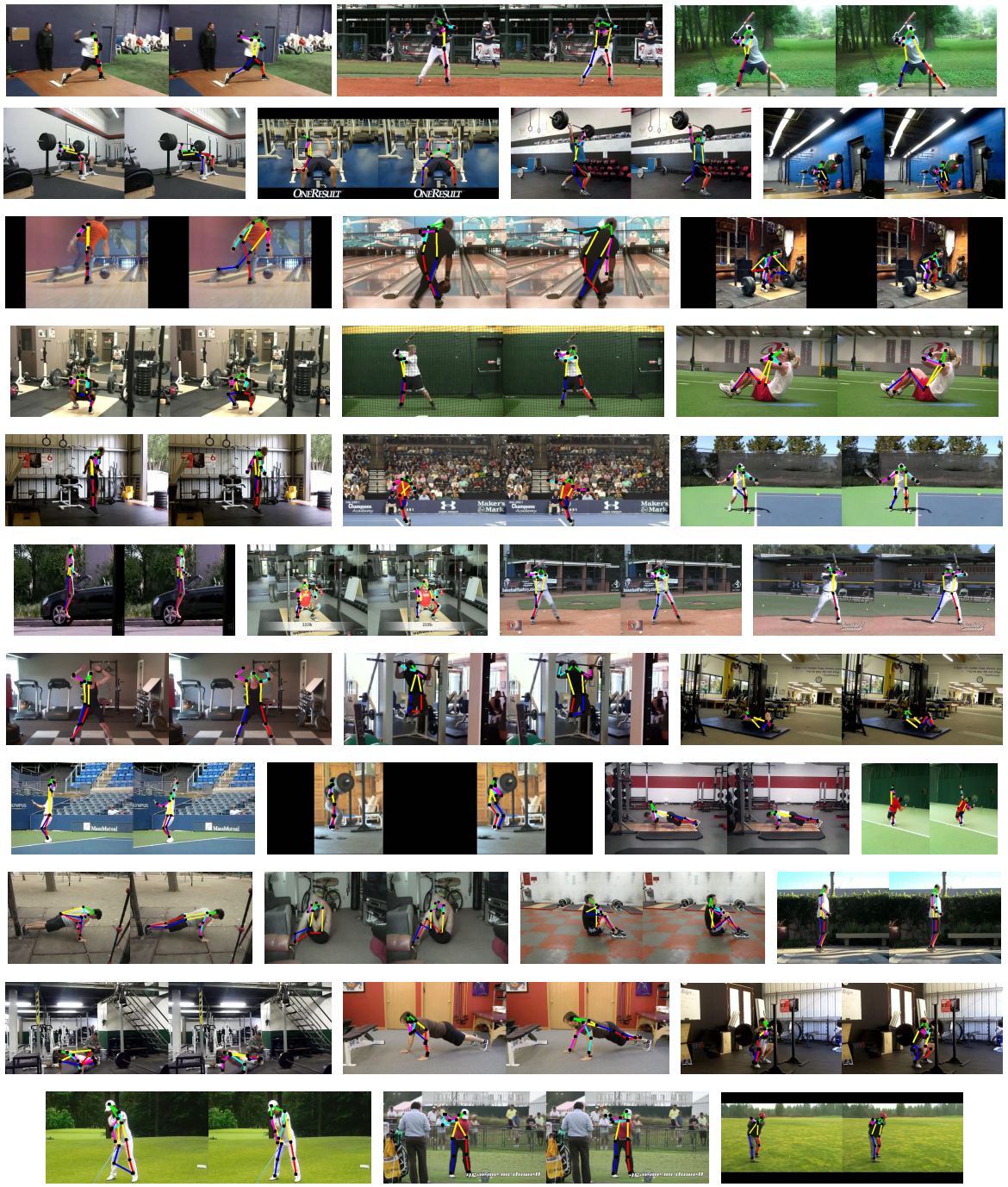


Figure 7.5: Qualitative results on some frames of the Penn-Action dataset as compared to our baseline with CCF. The left part of the images corresponds to the baseline while the right part shows improved poses obtained by the proposed ACPS.

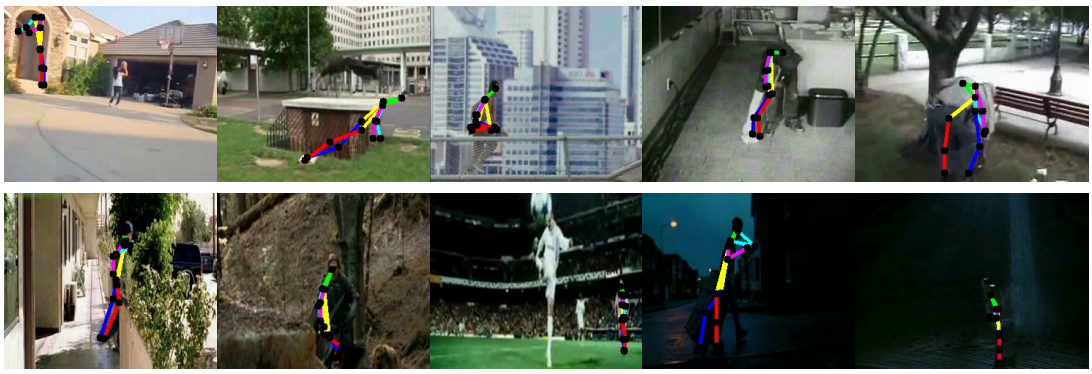


Figure 7.6: Few typical failure cases on the sub-J-HMDB due to large scale variations, rare poses with motion blur, large amount of body part occlusions, multiple persons, and bad illumination conditions.

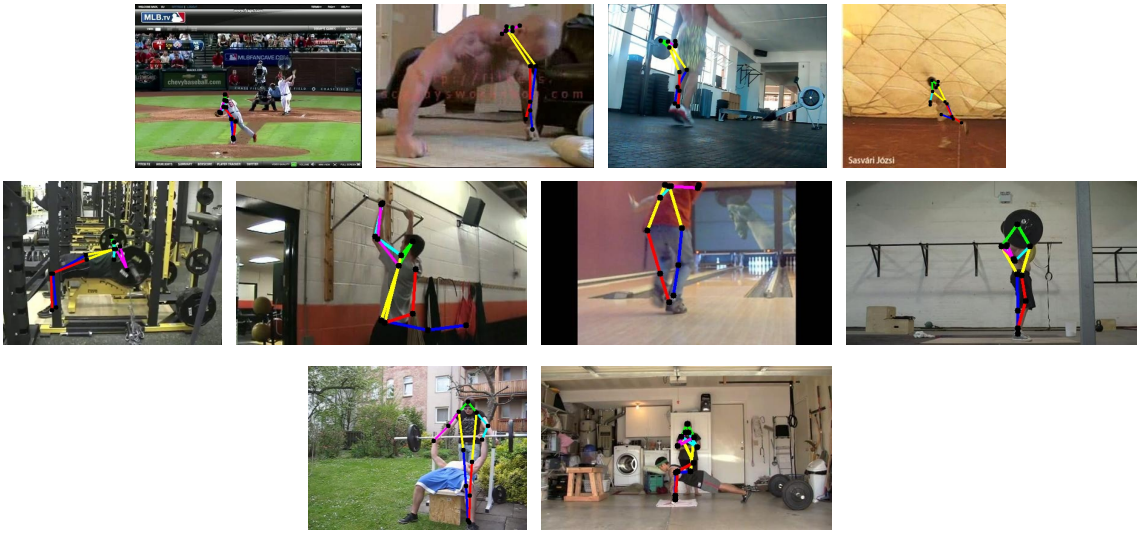


Figure 7.7: Few typical failure cases on the Penn-Action dataset due to large scale variations, motion blur, large amount of body part occlusions and truncations, and multiple persons.

CHAPTER 8

3D Pose Estimation from a Single Image

All of the methods presented so far in this thesis focus only on the estimation of 2D body pose. We demonstrated that the 2D body poses can be estimated reliably from unconstrained images and videos. However, many applications such as virtual reality or driver assistance systems also require the understanding of human body pose in 3D. Therefore, in this chapter we propose an approach for 3D human body pose estimation.

Existing methods for 3D pose estimation use deep neural networks to directly regress 3D pose from images. The main bottleneck between these approaches and their applicability in unconstrained environment is the availability of training data, since collecting large number of unconstrained training images with 3D pose annotations is practically infeasible. To address this shortcoming, in this chapter, we present a dual-source approach that does not require training data with 3D pose annotations, but instead relies on two independent sources of training data both of which are available in large numbers. The first source consists of accurate 3D motion capture data, and the second source consists of unconstrained images with annotated 2D poses. We combine both sources in a unified framework that first performs 2D pose estimation, as done in the previous chapters, and then lifts it to 3D using a robust approach for 3D pose reconstruction.

Contents

8.1. INTRODUCTION

3D human pose estimation has a vast range of applications such as virtual reality, human-computer interaction, activity recognition, sports video analytics, and autonomous vehicles. The problem has traditionally been tackled by utilizing multiple images captured by synchronized cameras capturing the person from multiple views (Belagiannis et al., 2014; Sigal et al., 2012; Yao et al., 2012a). In

many scenarios, however, capturing multiple views is infeasible which limits the applicability of such approaches. Since 3D human pose estimation from a single image is very difficult due to missing depth information, depth cameras have been utilized for human pose estimation (Baak et al., 2011; Shotton et al., 2011a; Grest et al., 2005). However, current depth sensors are also limited to indoor environments and cannot be used in unconstrained scenarios. Therefore, estimating 3D pose from single, in particular unconstrained, images is a highly relevant task.

One approach to address this problem is to follow a fully-supervised learning paradigm, where a regression model (Bo and Sminchisescu, 2010; Ionescu et al., 2014b; Kostrikov and Gall, 2014; Ionescu et al., 2014a; Agarwal and Triggs, 2006; Bo et al., 2008; Li and Chan, 2014; Tekin et al., 2016b) or a deep neural network (Li et al., 2015; Tekin et al., 2016a, 2017; Zhou et al., 2016b; Moreno-Noguer, 2017; Popa et al., 2017; Sun et al., 2017b; Pavlakos et al., 2017, 2018a) can be learned to directly regress the 3D pose from single images. This approach, however, requires a large amount of training data where each 2D image is annotated with a 3D pose. In contrast to 2D pose estimation, manual annotation of such training data is not possible due to ambiguous geometry and body part occlusions. On the other hand, automatic acquisition of accurate 3D pose for an image requires a very sophisticated setup. The popular datasets like HumanEva (Sigal et al., 2010) or Human3.6M (Ionescu et al., 2014b) use synchronized multiple cameras with a commercial marker-based system to acquire accurate 3D poses for images. This, however, requires a very expensive hardware setup and also limits the applicability of such systems primarily to indoor laboratory environments due to the requirements of marker-based system like studio environment and attached markers. Some recent approaches such as EgoCap (Rhodin et al., 2016) allows to capture 3D poses in outdoor environments, but image data in such cases is restricted only to ego-centric views of the person.

In this chapter, we propose a dual-source method that does not require training data consisting of pairs of an image and a 3D pose, but rather utilize 2D and 3D information from two independent training sources as illustrated in Figure 8.1. The first source is accurate 3D motion capture data containing a large number of 3D poses, and is captured in a laboratory setup, e.g., as in the CMU motion capture dataset (CMU, 2014) or the Human3.6M dataset (Ionescu et al., 2014b). Whereas, the second source consists of images with annotated 2D poses as they are provided by 2D human pose datasets, *e.g.*, MPII Human Pose (Andriluka et al., 2014), Leeds Sports Pose (Johnson and Everingham, 2010c), and MSCOCO (Lin et al., 2014a). Since 2D poses can be manually annotated for images, they do not impose any restriction regarding the environment from where the images are taken. In fact any image from the Internet can be annotated and used. Since both sources are captured independently, we do not know the 3D pose for any training image. In order to bring the two sources together, we map the motion capture data into a normalized 2D pose space to allow for an efficient retrieval based on 2D body joints. Concurrently, we learn a 2D pose estimation model from the 2D images based on convolutional neural networks. During inference, we first estimate the 2D pose and retrieve the nearest 3D poses using an effective approach that is robust to 2D pose estimation errors. We then jointly estimate the projection from the 3D pose space to the image and reconstruct the 3D pose. We extensively evaluate our approach on two popular datasets for 3D pose estimation namely Human3.6M (Ionescu et al., 2014b) and HumanEva (Sigal et al., 2010). We provide an in-depth analysis of the proposed approach. In particular, we analyze the impact of different MoCap datasets, the impact of the similarity of the training and test poses, the impact of the accuracy of the used 2D pose estimator, and also the differences of the skeleton structure between the two training

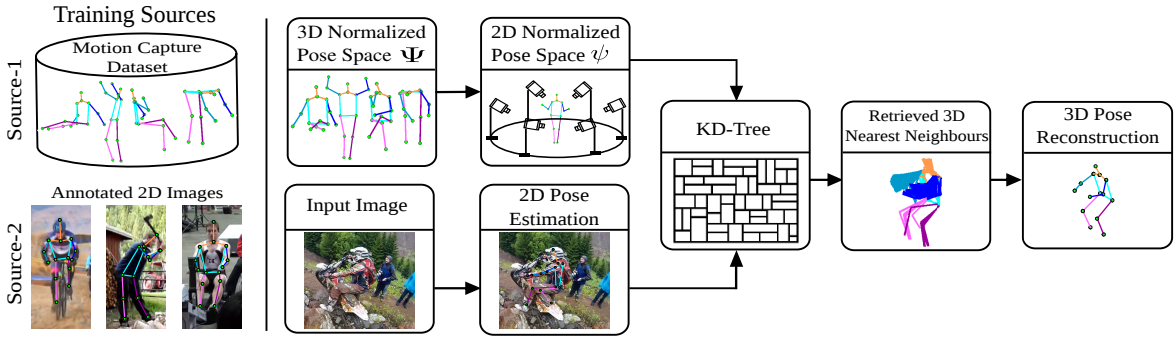


Figure 8.1: **Overview.** Our approach utilizes two training sources. The first source is a motion capture database that consists of only 3D poses. The second source is an image database with manually annotated 2D poses. The 3D poses in the motion capture data are normalized and projected to 2D using several virtual cameras. This gives many pairs of 3D-2D poses where the 2D poses are used as features for 3D pose retrieval. The image data is used to learn a 2D pose estimation model based on a CNN. Given a test image, the pose estimation model predicts the 2D pose which is then used to retrieve nearest 3D poses from the normalized 3D pose space. The final 3D pose is then estimated by minimizing the projection error under the constraint that the solution is close to the retrieved poses.

sources. Finally, we also provide qualitative results for images taken from the MPII Human Pose dataset (Andriluka et al., 2014).

8.2. OVERVIEW

In this chapter, we propose an approach to estimate the 3D pose from an RGB image. Since annotating 2D images with accurate 3D pose data is infeasible and obtaining 3D body pose data in unconstrained scenarios using sophisticated MoCap systems is impractical, our approach does not require that the training data consists of images annotated with 3D pose. In contrast, we use two independent sources of training data. The first source contains only 3D poses captured by a motion capture system. Such data is publicly available in large numbers and can also be captured in controlled indoor environments. The second source contains unconstrained images with annotated 2D poses, which are also abundantly available (Andriluka et al., 2014; Lin et al., 2014a) and can be easily annotated by humans. Apart from the requirement that the MoCap data contains poses that are related to the activities we are interested in, we do not assume any correspondence between the two sources. We therefore preprocess both sources separately as shown in Figure 8.1. From the image data, we learn a CNN based 2D pose estimation model to predict 2D poses from images. This will be described in Section 8.3. The MoCap data is processed to efficiently retrieve 3D poses that could correspond to a 2D pose. This part is discussed in Section 8.4.1. Finally, in Section 8.4.2, we estimate the 3D pose by minimizing the projection error under the constraint that the solution is close to the retrieved poses. The source code of the approach is publicly available.[1]

[1] http://pages.iai.uni-bonn.de/iqbal_umar/ds3dpose/

8.3. 2D POSE ESTIMATION

In this work, we use the convolutional pose machines (CPM) (Wei et al., 2016) for 2D pose estimation. Nevertheless, other CNN architectures, *e.g.*, stacked hourglass (Newell et al., 2016) or multi-context attention models (Chu et al., 2017a), could be used as well. Given an image I, we define the 2D pose of the person as $\mathbf{p} = \{\mathbf{x}_j\}_{j\in\mathcal{J}}$, where $\mathbf{x}_j = (x_j, y_j) \in \mathbb{R}^2$ denotes the 2D pixel coordinate of body joint j, and $\mathcal{J}$ is the set of all body joints. CPM consists of a multi-staged CNN architecture, where each stage $t \in \{1 \dots T\}$ produces a set of confidence scoremaps $\mathbf{H}_t = \{H_t^j\}_{j\in\mathcal{J}}$, where $H_t^j \in \mathbb{R}^{w\times h}$ is the confidence score map of body joint j at stage t, and w and h are the width and the height of the image, respectively. Each stage of the network sequentially refines the 2D pose estimates by utilizing the output of the preceding stage and also the features extracted from the raw input image. The final 2D pose $\mathbf{p}$ is obtained as

$$\mathbf{p} = \underset{\mathbf{p}'=\{\mathbf{x}'_j\}_{j\in\mathcal{J}}}{\arg\max} \sum_{j\in\mathcal{J}} H_T^j(\mathbf{x}'_j). \tag{8.1}$$

.

In our experiments we will show that training the network on publicly available dataset for 2D pose estimation in-the-wild, such as the MPII Human Pose dataset (Andriluka et al., 2014), is sufficient to obtain competitive results with our proposed method.

8.4. 3D POSE ESTIMATION

While the 2D pose estimation model is trained using the images annotated with 2D poses as shown in Figure 8.1, we now explain a method that utilizes the 3D poses from the second source to estimate the 3D pose from an image. Since both sources do not have any correspondence, we first have to establish correspondences between the 2D and 3D poses. For this, an estimated 2D pose is used as a query for 3D pose retrieval (Section 8.4.1). The retrieved 3D poses, however, contain many incorrect poses due to 2D-3D ambiguities, differences of the skeletons between the two training sources, and errors in the estimated 2D pose. It is therefore required to fit the 3D poses to the 2D observations. This is discussed in Section 8.4.2.

8.4.1. 3D Pose Retrieval

In order to efficiently retrieve 3D poses for a 2D pose query, we first preprocess the MoCap data by discarding the body location and orientation for each pose. This is achieved by applying the inverse transformation of the rigid transformation of the root joint, which is provided by the MoCap dataset, to all joints. After the transformation, the root joint is located at the origin of the coordinate system and the orientation of the pose is aligned with the x-axis. We denote the normalized 3D pose space with $\mathbf{\Omega}$, where $\mathbf{P} \in \mathbf{\Omega}$ denotes a normalized 3D pose. Similar to (Yasin et al., 2013), we project the normalized 3D poses $\mathbf{P} \in \mathbf{\Omega}$ to 2D using 120 virtual camera views with orthographic projection. We use elevation angles ranging between 0 and 60 degree and azimuth angles spanning 360 degrees, both sampled uniformly with a step size of 15 degrees. The projected 2D poses are further normalized by scaling such that the y-coordinates of the joints are within the range of $[-1, 1]$. The normalized 2D space does not depend on a specific coordinate system or a camera model and is denoted as ω. This step is illustrated in Figure 8.1. During inference, given a 2D pose estimated by the approach

explained in Section 8.3, we first normalize it according to ω, *i.e.*, we translate and scale the pose such that the y-coordinates of the joints are within the range of $[-1, 1]$. The normalized 2D pose is then used to retrieve 3D poses. We use the average Euclidean distance between the joint positions to measure the distance between two normalized 2D poses. Finally, we use a kd-tree (Krüger et al., 2010) to efficiently retrieve K-nearest neighbors in ω where the retrieved normalized 3D poses are the corresponding poses in $\mathbf{\Omega}$.

8.4.2. 3D Pose Estimation

In order to obtain the 3D pose $\mathbf{P} = \{\mathbf{X}_j\}_{j\in\mathcal{J}}$, we have to estimate the unknown projection $\mathcal{M}$ from the normalized pose space $\mathbf{\Omega}$ to the image. To this end, we minimize the energy

$$E(\mathbf{P}, \mathcal{M}) = E_p(\mathbf{P}, \mathcal{M}) + \alpha E_r(\mathbf{P}) \tag{8.2}$$

over $\mathbf{P}$ and $\mathcal{M}$. The parameter α defines the weighting between the two terms E_p and E_r.

The first term $E_p(\mathbf{P}, \mathcal{M})$ measures the projection error of the 3D pose $\mathbf{P}$ and the projection $\mathcal{M}$:

$$E_p(\mathbf{P}, \mathcal{M}) = \left(\sum_{j\in\mathcal{J}} \left\| \mathcal{M}\left(\mathbf{X}_j\right) - \mathbf{x}_j \right\|^2 \right)^{\frac{1}{2}}, \tag{8.3}$$

where $\mathbf{X}_j = (X, Y, Z)$ is the 3D position of the j^{th} joint of the unknown 3D pose $\mathbf{P}$ and $\mathbf{x}_j$ is the joint position of the predicted 2D pose $\mathbf{p}$.

The second term ensures that the pose $\mathbf{P}$ is close to the retrieved 3D poses $\mathbf{P}^{\mathsf{k}}$:

$$E_r(\mathbf{P}) = \sum_{\mathsf{k}} \left(\sum_{j\in\mathcal{J}} \left\| \mathbf{X}_j^{\mathsf{k}} - \mathbf{X}_j \right\|^2 \right)^{\frac{1}{2}}. \tag{8.4}$$

Minimizing the energy $E(\mathbf{P}, \mathcal{M})$ (8.2) over the continuous parameters $\mathcal{M}$ and $\mathbf{P}$ would be expensive. We therefore propose an approximate solution where we first estimate the projection $\mathcal{M}$ only. For the projection, we consider that the intrinsic parameters are provided and only estimate the global translation and orientation. The projection $\hat{\mathcal{M}}$ is estimated by minimizing

$$\hat{\mathcal{M}} = \arg\min_{\mathcal{M}} \left\{ \sum_{\mathsf{k}=1}^{\mathsf{K}} E_p(\mathbf{P}^{\mathsf{k}}, \mathcal{M}) \right\} \tag{8.5}$$

using non-linear gradient optimization with trust-region-reflective algorithm. We initialize the camera translation by $[0, 0, -Lf/l]$, where L is the mean height of the retrieved nearest neighbours and l corresponds to the height of the estimated 2D pose. In our experiments, we will also evaluate the case when the camera orientation and translation are also known. In this case, the projection $\mathcal{M}$ reduces to a rigid transformation of the 3D poses $\mathbf{P}$ from the normalized pose space $\mathbf{\Omega}$ to the camera coordinate system.

Given the estimated projection $\hat{\mathcal{M}}$, we minimize

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \left\{ E(\mathbf{P}, \hat{\mathcal{M}}) \right\} \tag{8.6}$$

to obtain the 3D pose $\mathbf{P}$.

The dimensionality of $\mathbf{P}$ can be reduced by applying PCA to the retrieved 3D poses $\mathbf{P}^{\mathsf{k}}$. Reducing the dimensions of $\mathbf{P}$ helps to decrease the optimization time without loss in accuracy, as we will show in the experiments.

8.5. EXPERIMENTS

We evaluate the proposed approach on two publicly available datasets, namely Human3.6M (Ionescu et al., 2014b) and HumanEva-I (Sigal et al., 2010). Both datasets provide accurate 3D poses for each image and camera parameters. For all cases, 2D pose estimation is performed by convolutional pose machines (Wei et al., 2016) trained on the MPII Human Pose dataset (Andriluka et al., 2014) without any fine-tuning, unless it is stated otherwise.

8.5.1. Evaluation on Human3.6M Dataset

For evaluation on the Human3.6M dataset, a number of protocols have been proposed in the literature. The protocol originally proposed for the Human3.6M dataset (Ionescu et al., 2014b), which we denote by *Protocol-III*, uses the annotated bounding boxes and the training data only from the action class of the test data. This simplifies the task due to the small pose variations for a single action class and the known person bounding box. Other protocols have been therefore proposed in (Kostrikov and Gall, 2014) and (Bogo et al., 2016). In order to compare with other existing approaches, we report results for all three protocols (Kostrikov and Gall, 2014; Bogo et al., 2016) and (Ionescu et al., 2014b).

8.5.1.1. Human3.6M Protocol-I

Protocol-I, which was proposed by Kostrikov and Gall (2014), is the most unconstrained protocol. It does not make any assumption about the location and activity labels during testing, and the training data comprises all action classes. The training set consists of six subjects (S1, S5, S6, S7, S8 and S9), whereas the testing is performed on every 64^{th} frame taken from the sequences of S11. For evaluation, we use the *3D pose error* as defined in (Simo-Serra et al., 2012). The error measures the accuracy of the relative pose up to a rigid transformation. To this end, the estimated skeleton is aligned to the ground-truth skeleton by a rigid transformation and the average 3D Euclidean joint error is measured after alignment. The body skeleton consists of 14 body joints namely head, neck, ankles, knees, hips, wrists, elbows, and shoulders. In order to comply with the protocol, we do not use any ground truth bounding boxes, but estimate them using an off-the-shelf person detector (Ren et al., 2015). The detected bounding boxes are used by the convolutional pose machines for 2D pose estimation. We consider two sources for the motion capture data, namely the Human3.6M and the CMU motion capture dataset.

We first evaluate the impact of the parameters of our approach and the impact of different MoCap datasets. We then compare our approach with the state-of-the-art and evaluate the impact of the 2D pose estimation accuracy.

Nearest Neighbors. The impact of the number of nearest neighbors K used during 3D pose reconstruction is evaluated in Figure 8.2. Increasing the number of nearest neighbors improves 3D pose estimation. This, however, also increases the reconstruction time. In the rest of this chapter, we use a default value of $K = 256$ that provides a good trade-off between accuracy and run-time. The reconstruction of the 3D pose with $K = 256$ for a single image takes roughly 0.6 seconds[2]. We

[2]Measured on a 3.4GHz Intel processor using only one core.

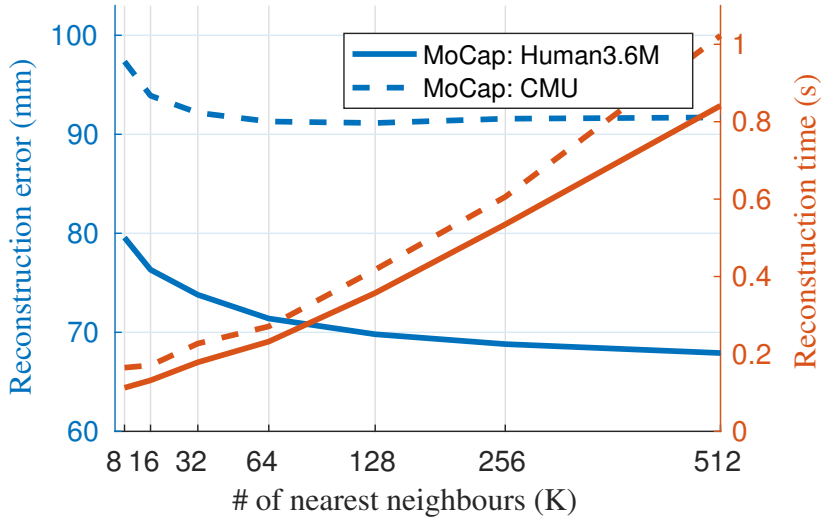


Figure 8.2: Impact of the number of nearest neighbors K.

can see that using the CMU MoCap dataset results in a higher error as compared to the Human3.6M dataset. We will evaluate the impact of different MoCap datasets in more details later in this section.

PCA. PCA can be used to reduce the dimension of $\mathbf{P}$. To this end, we use an adaptive approach and set the number of principal components based on the captured variance. The number of principal components therefore varies for each image. The impact of the threshold on the minimum amount of variation can be seen in Figure 8.3. If the threshold is within a reasonable range, *i.e.*, between 0.8 and 1, the accuracy is barely reduced while the runtime decreases significantly compared to 1, *i.e.*, without PCA. In this work, we use the minimum number of principle components that explain at least 80% of the variance of the retrieved 3D poses $\mathbf{P}^{\mathsf{k}}$.

Energy Terms. The impact of the weight α in (8.2) is reported in Figure 8.4. If $\alpha = 0$, the term E_r is ignored and the error is very high. This is expected since E_r constrains the possible solution while E_p ensures that the estimated 3D pose projects onto the estimated 2D pose. In our experiments, we use $\alpha = 1$.

Impact of MoCap dataset size. We evaluate the impact of the size of the MoCap dataset in Figure 8.5. In order to sub-sample the dataset, which consists of 469K 3D poses, we use a greedy approach that starts with an empty set and gradually adds a new pose if the distance to any previously selected pose is larger or equal to a threshold. Otherwise, the pose is discarded. Depending on the threshold (320mm, 160mm, 80mm, 40mm, 20mm), the dataset is reduced to 11K, 48K, 111K, 208K, and 329K poses, respectively. Using the entire 469K 3D poses of the Human3.6M training set as motion capture data results in a 3D pose error of 68.8mm. Reducing the size of the MoCap data to 329K reduces the error to 66.85mm. The reduction of the error is expected since the sub-sampling removes duplicates and very similar poses that do not provide any

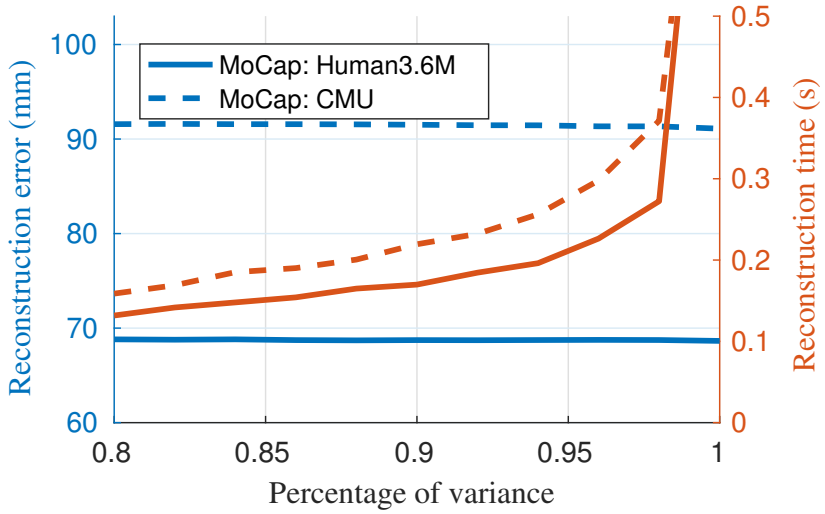


Figure 8.3: Impact of PCA. The number of principle components are selected based on the minimum number of components that explain a given percentage of variation. The x-axis corresponds to the threshold for the cumulative amount of variation.

additional information when they are retrieved. However, decreasing the size of the MoCap dataset even further degenerates the performance. In the rest of our experiments, we use the MoCap dataset from Human3.6M with 329K 3D poses, where a threshold of 20mm is used to remove similar poses. While the runtime of the approach is linear with respect to the number of nearest neighbors (K) as it can be observed in Figure 8.2, the sub-sampling of the MoCap dataset has a minimal impact on the runtime since the computational complexity of 3D pose retrieval is logarithmic with respect to the dataset size and the dataset size does not affect the energy function (8.2), in contrast to K.

CMU Motion Capture Dataset. Our approach does not require images that are annotated by 3D poses but uses MoCap data as a second training source. We therefore also evaluate the proposed method using the CMU MoCap dataset (CMU, 2014) to construct the 3D pose space. We down-sample the CMU dataset from 120Hz to 30Hz and use only one third of the 3D poses, resulting in 360K poses. We remove similar poses using the same threshold (20mm) as used for Human3.6M, which results in a final MoCap dataset with 303K 3D poses. Figure 8.6 compares the pose estimation accuracy using both datasets, while the results for each activity can be seen in Table 8.1. As expected the error is higher due to the differences of the datasets.

To analyze the impact of the MoCap data in more detail, we have evaluated the pose error for various modifications of the MoCap data in Table 8.1. First, we remove all poses of an activity from the MoCap data and evaluate the 3D pose error for the test images corresponding to the removed activity. The error increases since the dataset does not contain poses related to the removed activity anymore. While the error still stays comparable for many activities, *e.g.*Direction, Discussion, etc., a significant increase in error can be seen for activities that do not share similar poses with other activities, *e.g.*SitDown. However, even if all poses related to the activity of the test images are

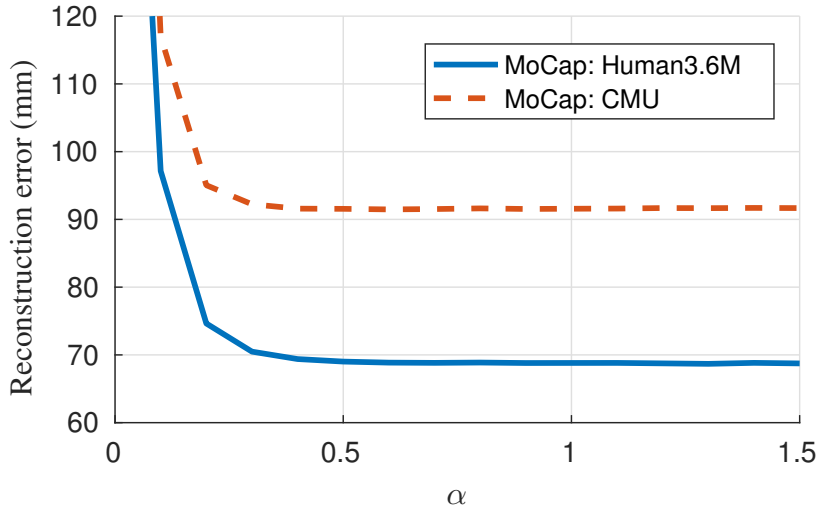


Figure 8.4: Impact of α.

MoCap data	Direction	Discuss	Eating	Greeting	Phoning	Posing	Purchases	Sit	SitDown
Human3.6M	59.5	52.4	75.5	67.0	58.8	64.9	58.2	68.4	89.7
Human3.6M \ Activity	61.2	52.3	92.6	70.2	61.1	66.5	59.3	85.6	122.2
Human3.6M ∈ Activity	68.8	57.6	70.8	73.7	62.9	66.7	63.4	73.4	99.4
Human3.6M + GT 3D Poses	52.9	45.7	59.9	60.1	50.4	54.1	51.6	56.3	71.7
CMU	73.3	64.7	95.9	80.2	85.7	81.8	77.1	110.5	138.8

MoCap data	Smoking	Photo	Waiting	Walk	WalkDog	WalkTogether	Mean	Median
Human3.6M	73.0	88.5	67.7	52.1	73.0	54.1	66.9	61.5
Human3.6M \ Activity	74.8	92.6	72.4	64.5	74.6	69.0	74.5	67.3
Human3.6M ∈ Activity	74.8	89.5	77.4	49.3	70.8	55.9	70.4	65.3
Human3.6M + GT 3D Poses	64.2	69.2	60.4	47.8	60.6	44.9	56.7	51.3
CMU	100.9	95.3	90.6	82.9	87.6	91.3	91.0	83.3

Table 8.1: Impact of the MoCap dataset. While for Human3.6M \ Activity we removed all poses from the dataset that correspond to the activity of the test sequence, Human3.6M ∈ Activity only contains the poses of the activity of the test sequence. For Human3.6M + GT 3D Poses, we include the ground-truth 3D poses of the test sequences to the MoCap dataset.

removed, the results are still good and better compared to the CMU dataset. This indicates that the error increase for the CMU dataset cannot only be explained by the difference of poses, but also other factors like different motion capture setups seem to influence the result. We will investigate the impact of the difference of the skeleton structure between two datasets in Section 8.5.2.

We also evaluate the case when the MoCap dataset contains only the poses of a specific activity. This also results in an increased mean pose estimation error and shows that having a diverse MoCap

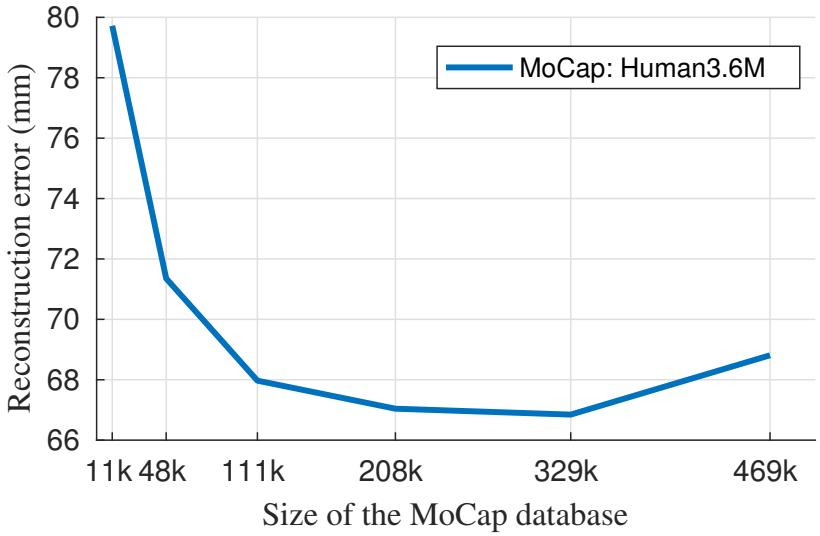


Figure 8.5: Impact of the size of the MoCap dataset.

dataset is helpful to obtain good performance. Finally, we also report the error when the 3D poses of the test sequences are added to the MoCap dataset. In this case, the mean error is reduced from 66.9mm to 56.7mm.

Comparison with State-of-the-art. Table 8.2 compares the performance of the proposed method with the state-of-the-art approaches (Kostrikov and Gall, 2014; Rogez and Schmid, 2016; Chen and Ramanan, 2017; Moreno-Noguer, 2017; Tome et al., 2017; Zhou et al., 2017a; Sun et al., 2017b) using both MoCap datasets. Our approach also outperforms the recent methods (Moreno-Noguer, 2017; Chen and Ramanan, 2017; Tome et al., 2017). While Moreno-Noguer (2017) utilizes 3D poses from Human3.6M as training data, Tome et al. (2017) use the 2D pose data from Human3.6M to learn a multistage deep CNN architecture for 2D pose estimation. We on the other hand do not use any 2D or 3D pose information for training and only utilize a pre-trained model trained on the MPII Human Pose Dataset (Andriluka et al., 2014) for 2D pose estimation. We also compare our performance with the most recent approaches (Zhou et al., 2017a; Sun et al., 2017b). These approaches perform better than our method. However, they use pairs of images and 3D poses to learn deep CNN models while our approach does not require 3D pose annotations for images. Moreover, in contrast to our method, none of the aforementioned approaches have shown that they can handle MoCap data that is from a different source than the test data.

Impact of 2D Pose. We also investigate the impact of the accuracy of the estimated 2D poses. If we initialize the approach with the 2D ground-truth poses, the 3D pose error is significantly reduced as shown in Table 8.3. This indicates that the 3D pose error can be further reduced by improving the used 2D pose estimation method. We also report the 3D pose error when both 3D and 2D ground-truth poses are available. In this case the error reduces even further which shows the potential of further improvements for the proposed method. We also compare our approach to (Chen and

Method	Direction	Discuss	Eating	Greeting	Phoning	Posing	Purchases	Sit	Sit Down
Kostrikov and Gall (2014)	-	-	-	-	-	-	-	-	-
Rogez and Schmid (2016)	-	-	-	-	-	-	-	-	-
Chen and Ramanan (2017)	71.6	66.6	74.7	79.1	70.1	67.6	89.3	90.7	195.6
Moreno-Noguer (2017)	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5
Tome et al. (2017)	-	-	-	-	-	-	-	-	-
Zhou et al. (2017a)	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1
Sun et al. (2017b)*	42.1	44.3	45.0	45.4	51.5	43.2	41.3	59.3	73.3
Ours	59.5	52.4	75.5	67.0	58.8	64.9	58.2	68.4	89.7
(MoCap from CMU dataset)									
Ours	73.3	64.7	95.9	80.2	85.7	81.8	77.1	110.5	138.8

Method	Smoking	Photo	Waiting	Walk	WalkDog	WalkTogehter	Mean	Median
Kostrikov and Gall (2014)	-	-	-	-	-	-	115.7	-
Rogez and Schmid (2016)	-	-	-	-	-	-	88.1	-
Chen and Ramanan (2017)	83.5	93.3	71.2	55.7	85.9	62.5	82.7	69.1
Moreno-Noguer (2017)	75.8	92.6	69.6	71.5	78.0	73.2	74.0	-
Tome et al. (2017)	-	-	-	-	-	-	70.7	-
Zhou et al. (2017a)	53.7	65.5	51.6	50.4	54.8	55.9	55.3	-
Sun et al. (2017b)*	51.0	53.0	44.0	38.3	48.0	44.8	48.3	-
Ours	73.0	88.5	67.7	52.1	73.0	54.1	66.9	61.5
(MoCap from CMU dataset)								
Ours	100.9	95.3	90.6	82.9	87.6	91.3	91.0	83.3

Table 8.2: Comparison with the state-of-the-art on the Human3.6M dataset using *Protocol-I*. *additional ground-truth information is used.

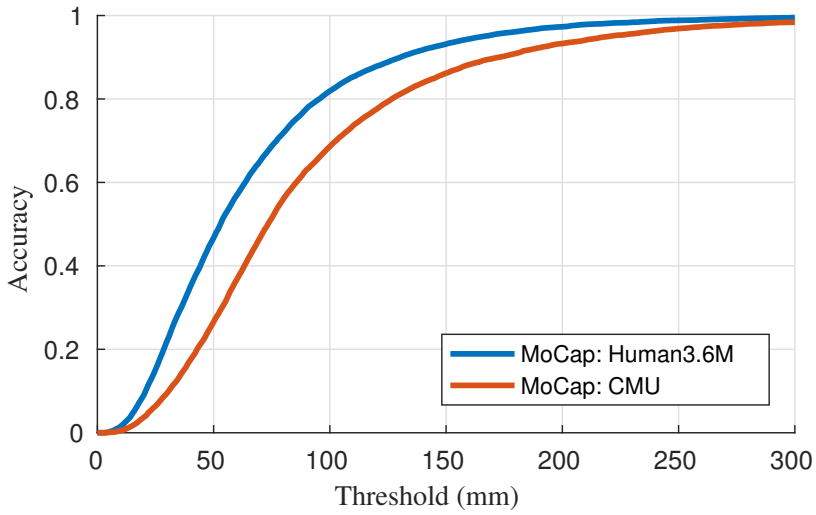


Figure 8.6: Comparison of 3D pose error using different MoCap datasets. The plot represent the percentage of estimated 3D poses with an error below a specific threshold.

Ramanan, 2017), which also report the accuracy for ground-truth 2D poses.

8.5.1.2. Human3.6M Protocol-II

The second protocol, *Protocol-II*, has been proposed in (Bogo et al., 2016). The dataset is split using five subjects (S1, S5, S6, S7, S8) for training and two subjects (S9 and S11) for testing. We follow (Lassner et al., 2017) and perform testing on every 5^{th} frame of the sequences from the frontal camera (cam-3) and trial-1 of each activity. The evaluation is performed in the same way as in *Protocol-I* with a body skeleton consisting of 14 joints. In contrast to *Protocol-I*, the *ground-truth bounding boxes* are, however, used during testing. Table 8.4 reports the comparison of the proposed method with the state-of-the-art approaches (Akhter and Black, 2015; Ramakrishna et al., 2012; Zhou et al., 2015; Bogo et al., 2016; Lassner et al., 2017; Tome et al., 2017; Moreno-Noguer, 2017; Martinez et al., 2017; Pavlakos et al., 2017; Tekin et al., 2017). While our approach achieves comparable results to (Akhter and Black, 2015; Ramakrishna et al., 2012; Zhou et al., 2015; Bogo et al., 2016; Lassner et al., 2017; Tome et al., 2017; Moreno-Noguer, 2017), more recent approaches (Martinez et al., 2017; Pavlakos et al., 2017; Tekin et al., 2017; Pavlakos et al., 2018a) perform better. The approaches (Pavlakos et al., 2017; Tekin et al., 2017), however, use pairs of images and 3D poses as training data, and the approach (Martinez et al., 2017) uses more recent improvements in the deep neural network architectures with exhaustive parameter selection to directly regress 3D pose from 2D joint information. Whereas, our approach does not require dataset specific training and therefore requires less supervision and can generalize better to different scenarios. The (Pavlakos et al., 2018a) is the current best performing approach, howoever, it utilizes additional annotations about the geometric relationships between body parts. We expect that using similar geometric relationships in our approach during 3D pose retrieval will also result in further improvements.

Method	Direction	Discuss	Eat	Greet	Phone	Pose	Purchase	Sit	SitDown
Ours	59.5	52.4	75.5	67.0	58.8	64.9	58.2	68.4	89.7
Ours + GT 2D	51.9	45.3	62.4	55.7	49.2	56.0	46.4	56.3	76.6
Ours + GT 2D + GT 3D	40.9	35.3	41.6	44.3	36.6	43.7	38.0	40.3	53.4
Chen and Ramanan (2017) + GT 2D	53.3	46.8	58.6	61.2	56.0	58.1	48.9	55.6	73.4
(MoCap from CMU dataset)									
Ours + GT 2D	67.8	58.7	90.3	72.1	78.2	75.7	71.9	103.2	132.8

Method	Smoke	Photo	Wait	Walk	WalkDog	WalkTogether	Mean	Median
Ours	73.0	88.5	67.7	52.1	73.0	54.1	66.9	61.5
Ours + GT 2D	58.8	79.1	58.9	35.6	63.4	46.3	56.1	51.9
Ours + GT 2D + GT 3D	44.2	56.6	45.9	26.9	45.8	31.4	41.6	39.1
Chen and Ramanan (2017) + GT 2D	60.3	76.1	62.2	35.8	61.9	51.1	57.5	51.9
(MoCap from CMU dataset)								
Ours + GT 2D	91.3	91.6	84.7	70.9	81.2	76.7	83.7	75.6

Table 8.3: Impact of the 2D pose estimation accuracy. GT 2D denotes that the ground-truth 2D pose is used. GT 3D denotes that the 3D poses of the test images are added to the MoCap dataset as in Table 8.1.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sit
Akhter and Black (2015)	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7
Ramakrishna et al. (2012)	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6
Zhou et al. (2015)	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1
Bogo et al. (2016)	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3
Moreno-Noguer (2017)	64.1	76.6	70.6	80.8	93.0	96.3	74.0	65.5	87.9
Lassner et al. (2017)	-	-	-	-		-	-	-	-
Tome et al. (2017)	-	-	-	-		-	-	-	-
Martinez et al. (2017)	44.8	52.0	44.4	50.5	61.7	59.4	45.1	41.9	66.3
Pavlakos et al. (2017)	-	-	-	-		-	-	-	-
Tekin et al. (2017)	-	-	-	-		-	-	-	-
Pavlakos et al. (2018a)	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7
Ours	75.3	75.8	70.9	92.8	89.0	101.5	78.1	61.4	97.9
(MoCap from CMU dataset)									
Ours	89.7	88.6	94.1	101.1	106.3	104.1	85.9	81.0	121.7

	SitDown	Smoking	Waiting	WalkDog	Walk	WalkTogether	Mean	Median
Akhter and Black (2015)	173.7	177.8	181.9	176.2	198.6	192.7	181.1	158.1
Ramakrishna et al. (2012)	175.6	160.4	161.7	150.0	174.8	150.2	157.3	136.8
Zhou et al. (2015)	137.5	106.0	102.2	106.5	110.4	115.2	106.7	90.0
Bogo et al. (2016)	137.3	83.4	77.3	79.7	86.8	81.7	82.3	69.3
Moreno-Noguer (2017)	109.5	83.8	93.1	81.6	73.5	72.6	81.5	-
Lassner et al. (2017)	-	-	-	-	-	-	80.7	-
Tome et al. (2017)	-	-	-	-	-	-	79.6	-
Martinez et al. (2017)	77.6	54.0	58.8	49.0	35.9	40.7	52.1	-
Pavlakos et al. (2017)	-	-	-	-	-	-	51.9	-
Tekin et al. (2017)	-	-	-	-	-	-	50.1	-
Pavlakos et al. (2018a)	56.8	42.6	39.6	43.9	32.1	36.5	41.8	-
Ours	121.6	84.2	85.8	75.8	67.8	65.0	83.8	75.3
(MoCap from CMU dataset)								
Ours	146.1	98.9	101.7	92.7	84.4	99.0	100.5	92.3

Table 8.4: Comparison with the state-of-the-art on the Human3.6M dataset using *Protocol-II*.

8.5.1.3. Human3.6M Protocol-III

The third protocol, *Protocol-III*, is the most commonly used protocol for Human3.6M. Similar to *Protocol-II*, the dataset is split by using subjects S1, S5, S6, S7 and S8 for training and subjects S9 and S11 for testing. The sequences are downsampled from the original frame-rate of 50fps to 10fps, and testing is performed on the sequences from all cameras and trials. The evaluation is performed without a rigid transformation, but both the ground-truth and estimated 3D poses are centered with respect to the root joint. We therefore have to use the provided camera parameters such that the estimated 3D pose is in the coordinate system of the camera. The training and testing is often performed on the same activity. However, some recent approaches also report results by training only once for all activities. In this work, we report results under both settings. In this protocol, a body skeleton with 17 joints is used and the ground-truth bounding boxes are used during testing. Note that even though the 3D poses contain 17 joints, we still use the 2D poses with 14 joints for nearest neighbor retrieval and only use the corresponding joints for optimizing objective (8.2). Table 8.5 provides a detailed comparison of the proposed approach with the state-of-the-art methods.

Finally, we present some qualitative results in Figure 8.7. As it can be seen, our approach shows very good performance even for highly articulated poses and under severe occlusions.

8.5.2. Evaluation on HumanEva-I Dataset

We follow the same protocol as described in (Simo-Serra et al., 2013; Kostrikov and Gall, 2014) and use the provided training data to train our approach while using the validation data as test set. As in (Simo-Serra et al., 2013; Kostrikov and Gall, 2014), we report our results on every 5^{th} frame of the sequences *walking* (A1) and *jogging* (A2) for all three subjects (S1, S2, S3) and camera C1. The 3D pose error is computed as in *Protocol-I* for the Human3.6M dataset.

We perform experiments with the 3D pose data from the HumanEva and CMU MoCap datasets. For HumanEva, we use the entire 49K 3D poses of the training data as MoCap dataset. Since the joint positions of the skeleton used for HumanEva differs from the joint annotations that are provided by the MPII Human Pose dataset, we fine-tune the 2D pose estimation model on the HumanEva dataset using the provided 2D pose data. For fine-tuning, we run 500 iterations with a learning rate of 0.00008.

We also have to adapt the skeleton structure of the CMU dataset to the skeleton structure of the HumanEva dataset. For this, we re-target the 3D poses in the CMU dataset to the skeleton of the HumaEva dataset using linear regression. For this, we first scale normalize the 3D poses in both datasets such that the height of each pose is equal to 1000mm. For each pose in the CMU dataset, we then search the nearest neighbor in the HumanEva dataset. For computing the distance between poses, we only use the joints that are common in both datasets. The pairs of poses that have a distance greater than 5mm are discarded and the remaining pairs are used to learn a linear mapping between the skeletons of the two datasets.

We analyze the impact of the difference between the skeletons of both datasets in Table 8.6. Using HumanEva as MoCap dataset results in a 3D pose error of 31.5mm, whereas using CMU as MoCap dataset increases the error significantly to 80.0mm. Re-targeting the skeletons of the CMU dataset to the skeleton of HumanEva reduces the error from 80.0mm to 50.5mm, and re-targeting the skeleton of HumanEva to CMU increases the error from 31.5mm to 58.4mm. This shows that the difference of the skeleton structure between the two sources can have a major impact on the evaluation. This is,

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases	Sit
Ionescu et al. (2014b)	132.7	183.6	132.4	164.4	162.1	205.9	150.6	171.3	151.6
Li and Chan (2014)	-	136.9	96.9	124.7	-	168.7	-	-	-
Tekin et al. (2016b)	102.4	158.5	88.0	126.8	118.4	185.0	114.7	107.6	136.2
Tekin et al. (2016a)	-	129.1	91.4	121.7	-	162.2	-	-	-
Du et al. (2016)	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5
Chen and Ramanan (2017)	89.9	97.6	90.0	107.9	107.3	139.2	93.6	136.1	133.1
Zhou et al. (2016a)	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5
Zhou et al. (2016b)	91.8	102.4	97.0	98.8	113.4	125.2	90.0	93.9	132.2
Sanzari et al. (2016)	48.8	56.3	96.0	84.8	96.5	105.6	66.3	107.4	116.9
Tome et al. (2017)	65.0	73.5	76.8	86.4	86.3	110.7	68.9	74.8	110.2
Rogez et al. (2017)	76.2	80.2	75.8	83.3	92.2	105.7	79.0	71.7	105.9
Moreno-Noguer (2017)	67.5	79.0	76.5	83.1	97.4	100.4	74.6	72.0	102.4
Mehta et al. (2017c)	62.6	78.1	63.4	72.5	88.3	93.8	63.1	74.8	106.6
Zhou et al. (2017a)	68.7	74.8	67.8	76.4	76.3	98.4	84.0	70.2	88.0
Mehta et al. (2016)	59.7	69.5	60.9	68.7	76.6	85.7	58.9	78.7	90.9
Lin et al. (2017a)	58.0	68.2	63.3	65.8	75.3	93.1	61.2	65.7	98.7
Pavlakos et al. (2017)	67.4	72.0	66.7	69.1	72	77.0	65.0	68.3	83.66
Tekin et al. (2017)	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1
Martinez et al. (2017)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0
Sun et al. (2017b)*	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7
Pavlakos et al. (2018a)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8
Ours	90.9	98.4	98.2	118.3	118.0	130.5	95.9	112.1	146.1
(MoCap from CMU dataset)									
Ours	139.4	148.0	148.3	165.2	161.7	170.1	138.6	168.2	168.5

	SitDown	Smoking	Waiting	WalkDog	Walk	WalkTogether	Mean	Median
Ionescu et al. (2014b)	243.0	162.1	170.7	177.1	96.6	127.9	162.1	-
Li and Chan (2014)	-	-	-	132.2	70.0	-	-	-
Tekin et al. (2016b)	205.7	118.2	146.7	128.1	65.9	77.2	125.3	-
Tekin et al. (2016a)	-	-	-	130.5	65.8	-	-	-
Du et al. (2016)	226.9	120.0	117.7	137.4	99.3	106.5	126.5	-
Chen and Ramanan (2017)	240.1	106.7	106.2	114.1	87.0	90.6	114.2	93.1
Zhou et al. (2016a)	199.2	107.4	118.1	114.2	79.4	97.7	113.0	-
Zhou et al. (2016b)	159.0	106.9	94.4	126.1	79.0	99.0	107.3	-
Sanzari et al. (2016)	129.6	97.8	65.9	130.5	92.6	102.2	93.2	-
Tome et al. (2017)	173.9	85.0	85.8	86.3	71.4	73.1	88.4	-
Rogez et al. (2017)	127.1	88.0	83.7	86.6	64.9	84.0	87.7	-
Moreno-Noguer (2017)	116.7	87.7	94.6	82.7	75.2	74.9	85.6	-
Mehta et al. (2017c)	138.7	78.8	73.9	82.0	55.8	59.6	80.5	-
Zhou et al. (2017a)	113.8	78.0	90.1	75.1	62.6	73.6	79.9	-
Mehta et al. (2016)	125.2	71.2	68.9	82.6	54.0	60.0	74.1	-
Lin et al. (2017a)	127.7	70.4	68.2	72.9	50.6	57.7	73.1	-
Pavlakos et al. (2017)	96.5	71.7	65.8	74.9	59.1	63.2	71.9	-
Tekin et al. (2017)	107.3	69.3	70.3	74.3‘	51.8	63.2	69.7	-
Martinez et al. (2017)	94.6	62.3	59.1	65.1	49.5	52.4	62.9	-
Sun et al. (2017b)*	86.7	61.5	53.4	61.6	47.1	53.4	59.1	-
(Pavlakos et al., 2018a)	71.1	56.6	52.9	60.9	44.7	47.8	56.2	-
Ours	150.1	112.4	113.5	109.2	89.1	88.4	111.8	95.3
(MoCap from CMU dataset)								
Ours	186.7	154.8	154.4	163.7	140.9	160.3	157.3	141.7

Table 8.5: Comparison with the state-of-the-art on the Human3.6M dataset using *Protocol-III*. *additional ground-truth information is used.

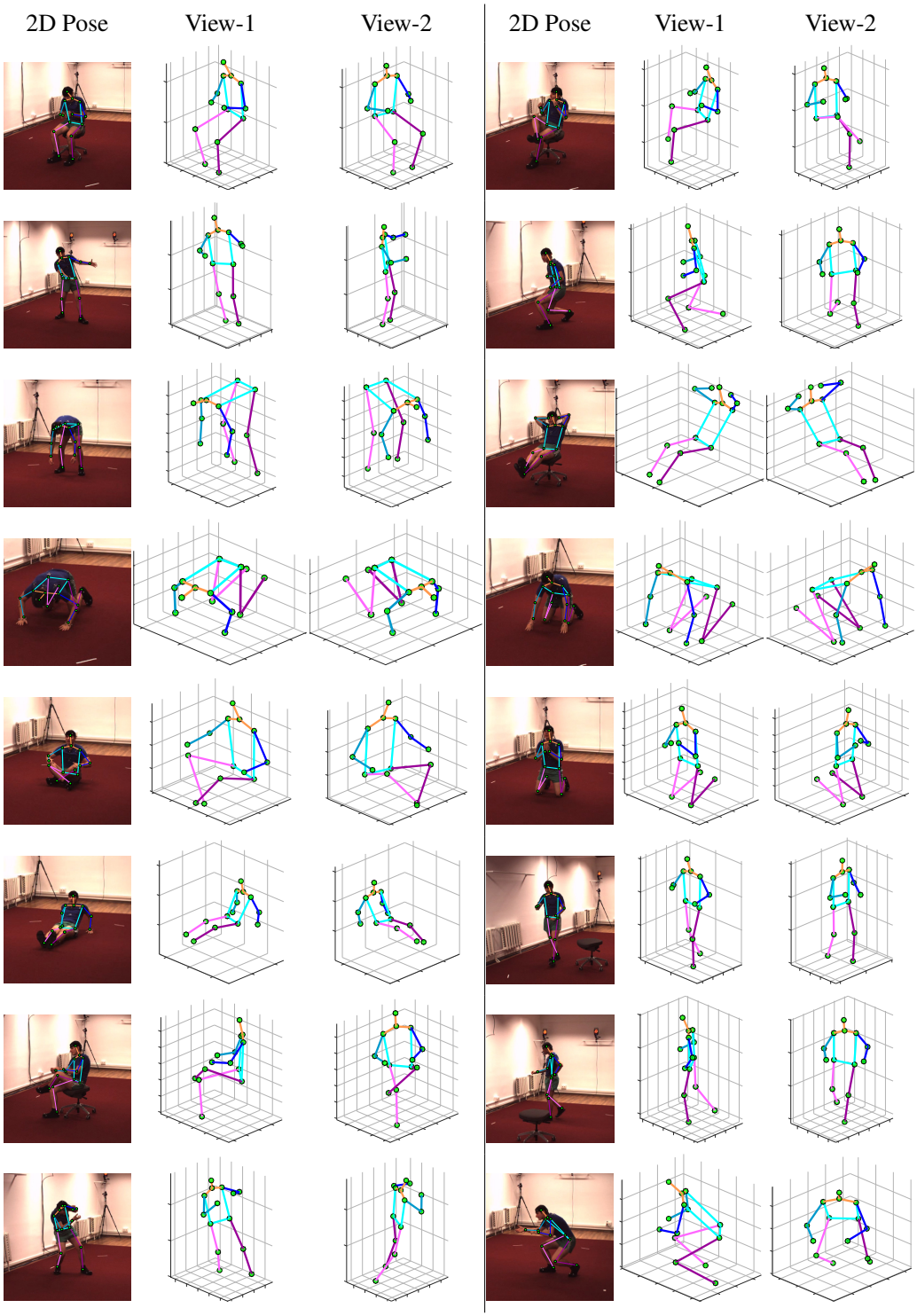


Figure 8.7: Some qualitative results from the Human3.6M (Ionescu et al., 2014b) dataset.

MoCap Data	Walking (A1, C1)			Jogging (A2, C1)			Average
	S1	S2	S3	S1	S2	S3	
HumanEva	27.4	28.6	32.5	39.9	29.4	31.4	31.5
CMU	68.4	81.6	88.3	70.1	81.6	89.9	80.0
CMU → HumanEva	39.5	47.3	61.4	53.5	48.3	53.1	50.5
HumanEva → CMU	45.1	54.9	59.1	58.6	63.1	69.7	58.4

Table 8.6: Impact of different skeleton structures. The symbol → indicates retargeting of the skeleton structure of one dataset to the skeleton of another dataset.

Methods	Walking (A1, C1)			Jogging (A2, C1)			Average
	S1	S2	S3	S1	S2	S3	
Simo-Serra et al. (2012)	99.6	108.3	127.4	109.2	93.1	115.8	108.9
Radwan et al. (2013)	75.1	99.8	93.8	79.2	89.8	99.4	89.5
Wang et al. (2014a)	71.9	75.7	85.3	62.6	77.7	54.4	71.3
Simo-Serra et al. (2013)	65.1	48.6	73.5	74.2	46.6	32.2	56.7
Kostrikov and Gall (2014)	44.0	30.9	41.7	57.2	35.0	33.3	40.3
Bo and Sminchisescu (2010)*	38.2	32.8	40.2	42.0	34.7	46.4	39.1
Lin et al. (2017a)	26.5	20.7	38.0	41.0	29.7	29.1	30.8
Popa et al. (2017)	27.1	18.4	39.5	37.6	28.9	27.6	29.9
Martinez et al. (2017)	19.7	17.4	46.8	26.9	18.2	18.6	24.6
Pavlakos et al. (2017)	22.3	19.5	29.7	28.9	21.9	23.8	24.3
Moreno-Noguer (2017)	19.8	12.6	26.2	43.8	21.8	22.1	24.4
Pavlakos et al. (2018a)	18.8	12.7	29.2	23.5	15.4	14.5	18.3
Ours	27.4	28.6	32.5	39.9	29.4	31.4	31.5
MoCap from CMU dataset							
Ours	39.5	47.3	61.4	53.5	48.3	53.1	50.5

Table 8.7: Comparison with other state-of-the-art approaches on the HumanEva-I dataset. The average 3D pose error (mm) is reported for all three subjects (S1, S2, S3) and camera C1. * denotes a different evaluation protocol.

however, not an issue for an application where the MoCap dataset defines the skeleton structure.

We also compare our approach with the state-of-the-art approaches (Kostrikov and Gall, 2014; Wang et al., 2014a; Radwan et al., 2013; Simo-Serra et al., 2013, 2012; Bo and Sminchisescu, 2010; Popa et al., 2017; Martinez et al., 2017; Pavlakos et al., 2017; Moreno-Noguer, 2017; Pavlakos et al., 2018a) in Table 8.7. Our method is competitive to all methods except of the very recent approaches (Moreno-Noguer, 2017; Martinez et al., 2017; Pavlakos et al., 2017, 2018a) that use more supervision or more recent CNN architectures. In particular, the ability to use MoCap data from a different source than the test data has so far not addressed by other works. This experimental protocol, however, is essential to assess the generalization capabilities of different methods.

Finally, we present qualitative results for a few realistic images taken from the MPII Human Pose dataset (Andriluka et al., 2014) in Figure 8.8. The results show that the proposed approach generalizes very well to complex unconstrained images.

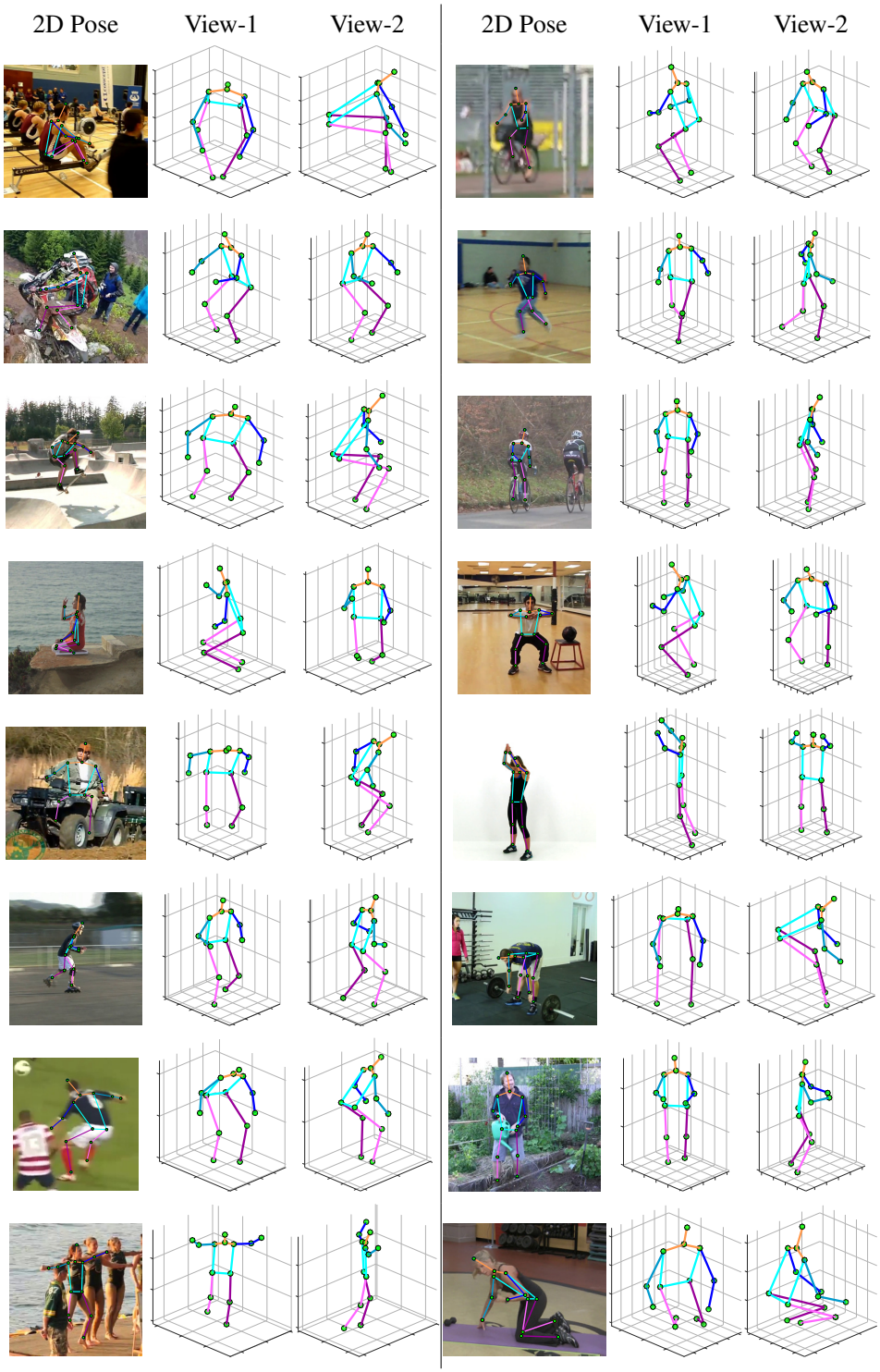


Figure 8.8: Some qualitative results from the MPII Human Pose Dataset.

8.6. SUMMARY

In this chapter, we have proposed a novel dual-source method for 3D human pose estimation from monocular images. The first source is a MoCap dataset with 3D poses and the other source are images with annotated 2D poses. Due to the separation of the two sources, our approach needs less supervision compared to approaches that are trained from images annotated with 3D poses, which is difficult to acquire under real conditions. The proposed approach therefore presents an important step towards accurate 3D pose estimation in unconstrained images. Compared to the preliminary work, the proposed approach does not require to train dataset specific models and can generalize across different scenarios. This is achieved by utilizing the strengths of recent 2D pose estimation methods and combining them with an efficient and robust method for 3D pose retrieval. We have performed a thorough experimental evaluation and demonstrated that our approach achieves competitive results in comparison to the state-of-the-art, even when the training data are from very different sources.

CHAPTER 9

Hand Pose Estimation via Latent 2.5D Heatmap Regression

All of the approaches presented so far in this thesis focus only on human body and completely ignore hands. Whereas, estimating the pose of a hand is an essential part of the human-computer interaction. Therefore, in this last chapter, we shift our attention to the challenging problem of hand pose estimation.

In the previous chapter, we presented an approach for 3D pose reconstruction from 2D poses. The approach has the advantage that it does not require training images with 3D pose annotation. However, relying only on 2D pose information has one major drawback that it can fall prey to re-projection ambiguities *e.g.*., the classic turning ballerina optical illusion. Therefore, it is essential to exploit the image information effectively to avoid resolve challenging ambiguous cases. To this end, in this chapter, we present a carefully designed novel 2.5D pose representation which brings-forth several advantages. First, it is invariant to scale and translation, therefore, can be estimated easily from an RGB image using a CNN. Second, it allows training the network in a multi-task setup hence, multiple sources of training data can be used. Third, and most importantly, it enables the exact recovery of the absolute 3D pose up to a scaling factor, where the scale can be estimated additionally given the prior of the hand size.

Contents

9.1. INTRODUCTION

Hand pose estimation from touch-less sensors enables advanced human machine interaction to increase comfort and safety. Estimating the pose accurately is a difficult task due to the large amounts of appearance variation, self occlusions and complexity of the articulated hand poses. 3D hand pose estimation escalates the difficulties even further since the depth of the hand keypoints also has to be estimated. To alleviate these challenges, many proposed solutions simplify the problem by using calibrated multi-view camera systems (Rehg and Kanade, 1994; de Campos and Murray, 2006; Oikonomidis et al., 2010; Rosales et al., 2001; Ballan et al., 2012; Sridhar et al., 2014; Tzionas et al., 2016; Panteleris and Argyros, 2017; Romero et al., 2017), depth sensors (Oikonomidis et al., 2011; Xu and Cheng, 2013; Qian et al., 2014; Taylor et al., 2014; Tang et al., 2014; Tompson et al., 2014a; Tang et al., 2015; Makris et al., 2015; Sridhar et al., 2015; Sun et al., 2015; Oberweger et al., 2015, 2016; Sridhar et al., 2016), or color markers/gloves (Wang and Popović, 2009). These approaches are, however, not very desirable due to their inapplicability in unconstrained environments. Therefore, in this work, we address the problem of 3D hand pose estimation from RGB images taken from the wild.

Given an RGB image of the hand, our goal is to estimate the 3D coordinates of hand keypoints relative to the camera. Estimating the 3D pose from a monocular hand image is an ill-posed problem due to scale and depth ambiguities. Attempting to do so will either not work at all, or results in overfitting to a very specific environment and subjects. We address these challenges by decomposing the problem into two subproblems both of which can be solved with significantly less ambiguity as compared to directly regressing the hand pose from an RGB image. To this end, we propose a novel 2.5D pose representation and then provide a solution to reconstruct the 3D pose from 2.5D. The proposed 2.5D representation is scale and translation invariant and can be easily estimated from RGB images. It consists of 2D coordinates of the hand keypoints in the input image and scale normalized depth for each keypoint relative to the root (palm). We perform scale normalization of the depth values such that one of the bones always has a fixed length in 3D space. Such a constrained normalization allows us to directly reconstruct the scale normalized absolute 3D pose. Our solution is still ill-posed because of relative normalized depth estimation, but it is better defined compared to relative or absolute depth estimation.

As a second contribution, we propose a novel CNN architecture to estimate the 2.5D pose from images. In the literature, there exist two main learning paradigms, namely heatmap regression (Wei et al., 2016; Newell et al., 2016) and holistic pose regression (Toshev and Szegedy, 2014; Sun et al., 2017b). Heatmap regression is now a standard approach for 2D pose estimation since it allows to accurately localize the keypoints in the image via per-pixel predictions. Creating volumetric heatmaps for 3D pose estimation (Pavlakos et al., 2017), however, results in very high computational overhead. Therefore, holistic regression is a standard approach for 3D pose estimation, but it suffers from accurate 2D keypoint localization. Since the 2.5D pose representation requires the prediction of both the 2D pose and depth values, we propose a new heatmap representation that we refer to as 2.5D heatmaps. It consists of 2D heatmaps for 2D keypoint localization and a depth map for each keypoint for depth prediction. We design the proposed CNN architecture such that the 2.5D heatmaps do not have to be designed by hand, but are learned in a latent way. We do this by a softargmax operation which converts the 2.5D heatmaps to 2.5D coordinates in a differentiable manner. The obtained 2.5D heatmaps are compact, invariant to scale and translation, and have the potential to localize keypoints

with sub-pixel accuracy.

The closest work to ours are the approaches of (Sun et al., 2017b; Zhou et al., 2017b; Pavlakos et al., 2017) in that they also perform 2.5D coordinate regression. While the approach in (Sun et al., 2017b) performs holistic pose regression with a fully connected output layer, (Zhou et al., 2017b) follows a hybrid approach and combines heatmap regression with holistic regression. Holistic regressions is shown to perform well for human body, but fails in cases where very precise localization is required, *e.g.*, fingertips in case of hands. In order to deal with this, the approach in (Pavlakos et al., 2017) performs dense volumetric regression. This, however, substantially increases the model size, which in turn forces to work at a lower spatial resolution. Our approach, on the other hand, retains the spatial resolution of the input and allows one to localize hand keypoints with sub-pixel accuracy. It enjoys the differentiability and compactness of holistic regression-based methods, translation invariance of volumetric representations, while also providing high spatial output resolution. Moreover, in contrast to existing methods, it does not require hand-designed target heatmaps, which can arguably be sub-optimal for a particular problem, but rather implicitly learns a latent 2.5D heatmap representation and converts them to 2.5D coordinates in a differentiable way.

Finally, note that given the 2.5D coordinates, the 3D pose has to be recovered. The existing approaches either make very strong assumptions such as the ground-truth location of the root (Sun et al., 2017b) and the global scale of the hand in 3D is known (Zhou et al., 2017b), or resort to an approximate solution (Pavlakos et al., 2017). The approach (Nie et al., 2017) tries to directly regress the absolute depth from the cropped and scaled image regions which is a very ambiguous task. In contrast, our approach does not make any assumptions, nor does it try to solve any ambiguous task. Instead, we propose a scale and translation invariant 2.5D pose representation, which can be easily obtained using CNNs, and then provide an exact solution to obtain the absolute 3D pose up to a scaling factor and only approximate the global scale of the hand.

We evaluate our approach on five challenging datasets with severe occlusions, hand object interactions and in-the-wild images. We demonstrate its effectiveness for both 2D and 3D hand pose estimation. The proposed approach outperforms state-of-the-art approaches by a large margin.

9.2. HAND POSE ESTIMATION

An overview of the proposed approach can be seen in Figure 9.1. Given an RGB image $\mathbf{I}$ of a hand, our goal is to estimate the 2D and 3D positions of all the $J = 21$ keypoints of the hand. We define the 2D hand pose as $\mathbf{p} = \{\mathbf{x}_j\}_{j \in \mathcal{J}}$ and 3D pose as $\mathbf{P} = \{\mathbf{X}_j\}_{j \in \mathcal{J}}$, where $\mathbf{x}_j = (x_j, y_j) \in \mathbb{R}^2$ represents the 2D pixel coordinates of the keypoint j in image $\mathbf{I}$ and $\mathbf{X}_j = (X_j, Y_j, Z_j) \in \mathbb{R}^3$ denotes the location of the keypoint in the 3D camera coordinate frame measured in millimeters. The Z-axis corresponds to the optical axis. Given the intrinsic camera parameters $\mathcal{K}$, the relationship between the 3D location $\mathbf{X}_j$ and corresponding 2D projection $\mathbf{x}_j$ can be written as follows under a perspective projection:

$$Z_j \begin{pmatrix} x_j \\ y_j \\ 1 \end{pmatrix} = \mathcal{K} \begin{pmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{pmatrix} = \mathcal{K} \begin{pmatrix} X_j \\ Y_j \\ Z_{root} + Z_j^r \\ 1 \end{pmatrix} \quad j \in \mathcal{J} \tag{9.1}$$

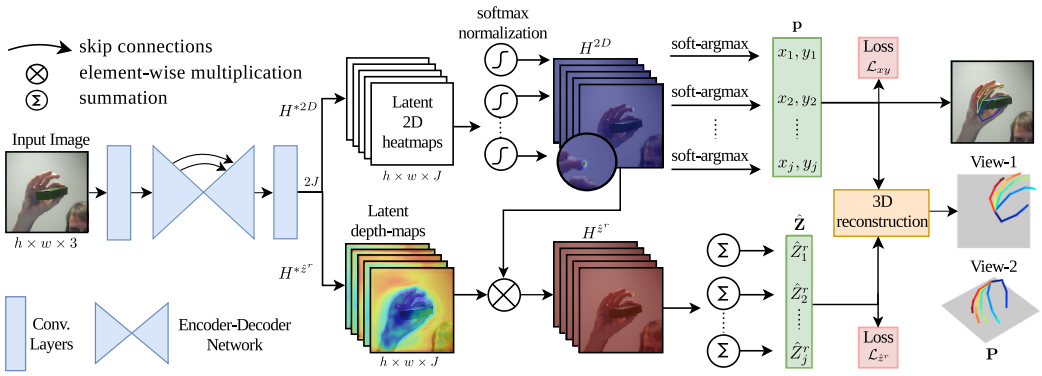


Figure 9.1: Overview of the proposed approach. Given an image of a hand, the proposed CNN architecture produces latent 2.5D heatmaps containing the latent 2D heatmaps H^{*2D} and latent depth maps $H^{*\hat{z}}$. The latent 2D heatmaps are converted to probability maps H^{2D} using softmax normalization. The depth maps $H^{\hat{z}}$ are obtained by multiplying the latent depth maps $H^{*\hat{z}}$ with the 2D heatmaps. The 2D pose $\mathbf{p}$ is obtained by applying spatial softargmax on the 2D heatmaps, whereas the normalized depth values $\hat{\mathbf{Z}}^r$ are obtained by the summation of depth maps. The final 3D pose is then estimated by the proposed approach for reconstructing 3D pose from 2.5D.

where $j \in \mathcal{J}$, Z_{root} is the depth of the root keypoint, and $Z_j^r = Z_j - Z_{root}$ corresponds to the depth of the j^{th} keypoint relative to the root. In this work we use the palm of the hand as the root keypoint.

9.2.1. The 2.5D Pose Representation

Given an image $\mathbf{I}$, we need to have a function $\mathcal{F}$, such that $\mathcal{F} : \mathbf{I} \rightarrow \mathbf{P}$, and the estimated 3D hand pose $\mathbf{P}$ can be projected to 2D with the camera parameters $\mathcal{K}$. However, predicting the absolute 3D hand pose in camera coordinates is infeasible due to irreversible geometry and scale ambiguities. We, therefore, choose a 2.5D pose representation, which is much easier to be recovered from a 2D image, and provide a solution to recover the 3D pose from the 2.5D representation. We define the 2.5D pose as $\mathbf{P}^{2.5D} = \{\mathbf{X}_j^{2.5D}\}_{j \in \mathcal{J}}$, where $\mathbf{X}_j^{2.5D} = (x_j, y_j, Z_j^r)$. The coordinates x_j and y_j are the image pixel coordinates of the j^{th} keypoint and Z_j^r is its metric depth relative to the root keypoint. Moreover, in order to remove the scale ambiguities, we scale-normalize the 3D pose as follows:

$$\hat{\mathbf{P}} = \frac{C}{s} \cdot \mathbf{P}, \tag{9.2}$$

where $s = \|\mathbf{X}_n - \mathbf{X}_{parent(n)}\|_2$ is computed for each 3D pose independently. This results in a normalized 3D pose $\hat{\mathbf{P}}$ with a constant distance C between a specific pair of keypoints $(n, parent(n))$. Subsequently, our normalized 2.5D representation for keypoint j becomes $\hat{\mathbf{X}}_j^{2.5D} = (x_j, y_j, \hat{Z}_j^r)$. Note that the 2D pose does not change due to the normalization, since the projection of the 3D pose remains the same. Such a normalized 2.5D representation has several advantages: it allows to effectively exploit image information, it enables dense pixel-wise prediction (Section 9.3), it allows us to perform multi-task learning so that multiple sources of training data can be used, and finally it allows us to devise an approach to exactly recover the absolute 3D pose up to a scaling factor. We describe the proposed solution to obtain the function $\mathcal{F}$ in Section 9.3, while the 3D pose reconstruction from 2.5D pose is explained in the next section.

9.2.2. 3D Pose Reconstruction from 2.5D

Given a 2.5D pose $\hat{\mathbf{P}}^{2.5D} = \mathcal{F}(\mathbf{I})$, we need to find the depth $\hat{Z}_{root}$ of the root keypoint to reconstruct the scale normalized 3D pose $\hat{\mathbf{P}}$ using (9.1). While there exist many 3D poses that can have the same 2D projection, given the 2.5D pose and intrinsic camera parameters, there exists a unique 3D pose that satisfies

$$(\hat{X}_n - \hat{X}_m)^2 + (\hat{Y}_n - \hat{Y}_m)^2 + (\hat{Z}_n - \hat{Z}_m)^2 = C^2, \tag{9.3}$$

where $(n,\ m = parent(n))$ is the pair of keypoints used for normalization in (9.2). The equation above can be rewritten in terms of the 2D projections (x_n, y_n) and (x_m, y_m) as follows:

$$(x_n\hat{Z}_n - x_m\hat{Z}_m)^2 + (y_n\hat{Z}_n - y_m\hat{Z}_m)^2 + (\hat{Z}_n - \hat{Z}_m)^2 = C^2. \tag{9.4}$$

Subsequently, replacing $\hat{Z}_n$ and $\hat{Z}_m$ with $(\hat{Z}_{root} + \hat{Z}^r_n)$ and $(\hat{Z}_{root} + \hat{Z}^r_m)$, respectively, yields:

$$\begin{aligned}(x_n(\hat{Z}_{root} + \hat{Z}^r_n) - x_m(\hat{Z}_{root} + \hat{Z}^r_m))^2 + (y_n(\hat{Z}_{root} + \hat{Z}^r_n) - y_m(\hat{Z}_{root} + \hat{Z}^r_m))^2 \\ + ((\hat{Z}_{root} + \hat{Z}^r_n) - (\hat{Z}_{root} + \hat{Z}^r_m))^2 = C^2.\end{aligned} \tag{9.5}$$

Given the 2.5D coordinates of both keypoints n and m, Z_{root} is the only unknown in the equation above. Simplifying the equation further leads to a quadratic equation with the following coefficients

$$\begin{aligned}a &= (x_n - x_m)^2 + (y_n - y_m)^2 \\ b &= \hat{Z}^r_n(x_n^2 + y_n^2 - x_nx_m - y_ny_m) + \hat{Z}^r_m(x_m^2 + y_m^2 - x_nx_m - y_ny_m) \\ c &= (x_n\hat{Z}^r_n - x_m\hat{Z}^r_m)^2 + (y_n\hat{Z}^r_n - y_m\hat{Z}^r_m)^2 + (\hat{Z}^r_n - \hat{Z}^r_m)^2 - C^2.\end{aligned} \tag{9.6}$$

This results in two values for the unknown variable Z_{root}, one in front of the camera and one behind the camera. We choose the solution in front of the camera

$$\hat{Z}_{root} = 0.5(-b + \sqrt{b^2 - 4ac})/a. \tag{9.7}$$

Given the value of Z_{root}, $\hat{\mathbf{P}}^{2.5D}$, and the intrinsic camera parameters $\mathcal{K}$, the scale normalized 3D pose can be reconstructed by back-projecting the 2D pose $\mathbf{p}$ using (9.1). In this work, we use $C = 1$ and the distance between the first joint (metacarpophalangeal - MCP) of the index finger and the palm (root) to calculate the scaling factor s. We choose these keypoints since they are the most stable in terms of 2D pose estimation.

9.2.3. Scale Recovery

Up to this point, we have obtained the 2D and scale normalized 3D pose $\hat{\mathbf{P}} = \{\hat{\mathbf{X}}_j\}_{j\in\mathcal{J}}$ of the hand, where $\hat{\mathbf{X}}_j$ denotes the scale-normalized 3D location of the keypoint j. In order to recover the absolute 3D pose $\mathbf{P}$, we need to know the global scale of the hand. In many scenarios this can be known a priori, however, in case it is not available, we estimate the scale $\hat{s}$ by

$$\hat{s} = \underset{s}{\operatorname{argmin}} \sum_{j,k\in\mathcal{E}} (s \cdot \|\hat{\mathbf{X}}_j - \hat{\mathbf{X}}_k\| - \mu_{jk})^2, \tag{9.8}$$

where μ_{jk} is the mean length of the bone between keypoints j and k in the training data, and $\mathcal{E}$ defines the kinematic structure of the hand.

9.3. 2.5D POSE REGRESSION

In order to regress the 2.5D pose $\hat{\mathbf{P}}^{2.5D}$ from an RGB image of the hand, we learn the function $\mathcal{F}$ using a CNN. In this section, we first describe an alternative formulation of the CNN (Section 9.3.1) and then describe our proposed solution for regressing latent 2.5D heatmaps in Section 9.3.2. In all formulations, we train the CNNs using a loss function $\mathcal{L}$ which consists of two parts $\mathcal{L}_{xy}$ and $\mathcal{L}_{\hat{Z}^r}$, each responsible for the regression of 2D pose and root-relative depths for the hand keypoints, respectively. Formally, the loss can be written as follows:

$$\mathcal{L}(\hat{\mathbf{P}}^{2.5D}) = \mathcal{L}_{xy}(\mathbf{p}, \mathbf{p}_{gt}) + \alpha \mathcal{L}_{\hat{Z}^r}(\hat{\mathbf{Z}}^r, \hat{\mathbf{Z}}^{r,gt}), \tag{9.9}$$

where $\hat{\mathbf{Z}}^r = \{\hat{Z}^r_j\}_{j\in\mathcal{J}}$ and $\hat{\mathbf{Z}}^{r,gt} = \{\hat{Z}^{r,gt}_j\}_{j\in\mathcal{J}}$ and gt refers to ground-truth annotations. This loss function has the advantage that it allows to utilize multiple sources of training, *i.e.*, in-the-wild images with only 2D pose annotations and constrained or synthetic images with accurate 3D pose annotations. While $\mathcal{L}_{xy}$ is valid for all training samples, $\mathcal{L}_{\hat{Z}^r}$ is enforced only when the 3D pose annotations are available, otherwise it is not considered.

9.3.1. Direct 2.5D Heatmap Regression

Heatmap regression is the de-facto approach for 2D pose estimation (Tompson et al., 2015; Wei et al., 2016; Newell et al., 2016; Simon et al., 2017). In contrast to holistic regression, heatmaps have the advantage of providing higher output resolution, which helps in accurately localizing the keypoints. However, they are scarcely used for 3D pose estimation since a 3D volumetric heatmap representation (Pavlakos et al., 2017) results in a high computational and storage cost.

We, thus, propose a novel and compact heatmap representation, which we refer to as 2.5D heatmaps. It consists of 2D heatmaps H^{2D} for keypoint localization and depth maps $H^{\hat{z}^r}$ for depth predictions. While the 2D heatmap H^{2D}_j represents the likelihood of the j^{th} keypoint at each pixel location, the depth map $H^{\hat{z}^r}_j$ provides the scale normalized and root-relative depth prediction for the corresponding pixels. This representation of depth maps is scale and translation invariant and remains consistent across similar hand poses, therefore, it is significantly easier to be learned using CNNs. The CNN provides a $2J$ channel output with J channels for 2D localization heatmaps H^{2D} and J channels for depth maps $H^{\hat{z}^r}$. The target heatmap $H^{2D,gt}_j$ for the j^{th} keypoint is defined as

$$H^{2D,gt}_j(p) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}^{gt}_j\|}{\sigma^2}\right), \quad \mathbf{x} \in \mathcal{X} \tag{9.10}$$

where $\mathbf{x}^{gt}_j$ is the ground-truth 2D location of the j^{th} keypoint, σ controls the standard deviation of the heatmaps and $\mathcal{X}$ is the set of all pixel locations in image $\mathbf{I}$. Since the ground-truth depth maps are not available, we define them by

$$H^{\hat{z}^r}_j = \hat{Z}^{r,gt}_j \cdot H^{2D,gt}_j \tag{9.11}$$

where $\hat{Z}^{r,gt}_j$ is the ground-truth normalized root-relative depth value of the j^{th} keypoint. During inference, the 2D keypoint position is obtained as the pixel with the maximum likelihood

$$\mathbf{x}_j = \underset{\mathbf{x}}{\operatorname{argmax}}\, H^{2D}_j(\mathbf{x}), \tag{9.12}$$

and the corresponding depth value is obtained as,

$$\hat{Z}_j^r = H_j^{\hat{z}^r}(\mathbf{x}_j). \tag{9.13}$$

9.3.2. Latent 2.5D Heatmap Regression

The 2.5D heatmap representation as described in the previous section is, arguably, not the most optimal representation. First, the ground-truth heatmaps are hand designed and are not ideal, *i.e.*, σ remains fixed for all keypoints and cannot be learned due to indifferentiability of (9.12). Ideally, it should be adapted for each keypoint, *e.g.*, heatmaps should be very peaky for fingertips while relatively wide for the palm. Secondly, the Gaussian distribution is a natural choice for 2D keypoint localization, but it is not very intuitive for depth prediction, *i.e.*, the depth stays roughly the same throughout the palm but is modeled as Gaussians. Therefore, we alleviate these problems by proposing a latent representation of 2.5D heatmaps, *i.e.*, the CNN learns the optimal representation by minimizing a loss function in a differentiable way.

To this end, we consider the $2J$ channel output of the CNN as latent variables H_j^{*2D} and $H_j^{*\hat{z}^r}$ for 2D heatmaps and depth maps, respectively. We then apply spatial softmax normalization to the 2D heatmap H_j^{*2D} of each keypoint j to convert it to a probability map

$$H_k^{2D}(\mathbf{x}) = \frac{\exp(\beta_j H_j^{*2D}(\mathbf{x}))}{\sum_{\mathbf{x}' \in \mathcal{X}} \exp(\beta_j H_j^{*2D}(\mathbf{x}'))}, \tag{9.14}$$

where $\mathcal{X}$ is the set of all pixel locations in the input map H_j^{*2D}, and β_j is the learnable parameter that controls the spread of the output heatmaps H^{2D}. Finally, the 2D keypoint position of the j^{th} keypoint is obtained as the weighted average of the 2D pixel coordinates as,

$$\mathbf{x}_j = \sum_{\mathbf{x} \in \mathcal{X}} H_j^{2D}(\mathbf{x}) \cdot \mathbf{x}, \tag{9.15}$$

while the corresponding depth value is obtained as the summation of the Hadamard product of $H_j^{2D}(\mathbf{x})$ and $H_j^{*\hat{z}^r}(\mathbf{x})$ as follows

$$\hat{Z}_j^r = \sum_{\mathbf{x} \in \mathcal{X}} H_j^{2D}(\mathbf{x}) \circ H_j^{*\hat{z}^r}(\mathbf{x}). \tag{9.16}$$

A pictorial representation of this process can be seen in Figure 9.1. The operation in (9.15) is known as softargmax in the literature (Chapelle and Wu, 2010). Note that the computation of both the 2D keypoint location and the corresponding depth value is fully differentiable. Hence the network can be trained end-to-end, while generating a latent 2.5D heatmap representation. In contrast to the heatmaps with fixed standard deviation in Section 9.3.1, the spread of the latent heatmaps can be adapted for each keypoint by learning the parameter β_j, while the depth maps are also learned implicitly without any ad-hoc design choices. A comparison between heatmaps obtained by direct heatmap regression and the ones implicitly learned by latent heatmap regression can be seen in Figure 9.2.

9.4. EXPERIMENTS

In this section, we evaluate the performance of the proposed approach in detail and also compare it with the state-of-the-art. For this, we use five challenging datasets – namely, the Dexter+Object

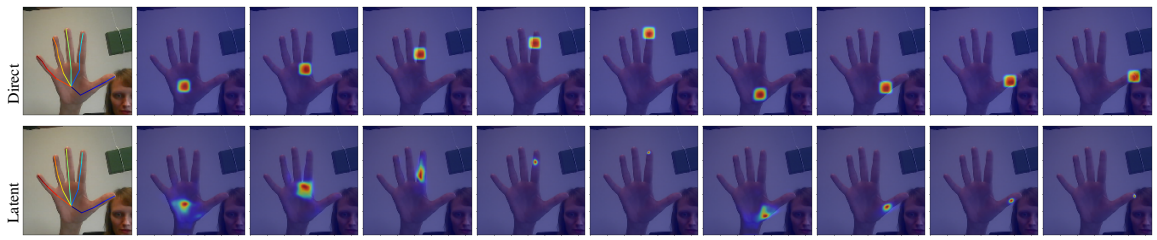


Figure 9.2: Comparison between the heatmaps obtained using direct heatmap regression (Section 9.3.1) and the proposed latent heatmap regression approach (Section 9.3.2). We can see how the proposed method automatically learns the spread separately for each keypoint, *i.e.*, very peaky for fingertips while a bit wider for the palm.

dataset (Sridhar et al., 2016), the Ego-Dexter dataset (Mueller et al., 2017), the Stereo Hand Pose dataset (Zhang et al., 2016) dataset, the Rendered Hand Pose dataset (Zimmermann and Brox, 2017), and the MPII+NZSL dataset (Simon et al., 2017). The details of each dataset are as follows.

Dexter+Object (D+O). The D+O dataset (Sridhar et al., 2016) provides 6 test video sequences with 3145 frames in total. All sequences are recorded using a static camera with a single person interacting with an object. The dataset provides both 2D and 3D pose annotations for the fingertips of the left hand.

EgoDexter (ED). The ED dataset (Mueller et al., 2017) provides both 2D and 3D pose annotations for 4 testing video sequences with 3190 frames. The videos are recorded with body-mounted camera from egocentric viewpoints and contains cluttered backgrounds, fast camera motion, and complex interactions with various objects. Similar to D+O dataset, it only provides annotations for the fingertips. In addition, (Mueller et al., 2017) also provides the so called *SynthHands* dataset containing synthetic images of hands from ego-centric views with accurate 3D pose annotations. The images are provided with chroma-keyed background, that we replace with random backgrounds from NYU depth dataset (Nathan Silberman and Fergus, 2012) and use them as additional training data for testing on the ED dataset.

Stereo Hand Pose (SHP). The SHP dataset (Zhang et al., 2016) provides 2D and 3D pose annotations of 21 keypoints for 6 pairs of stereo sequences with a total of 18000 stereo pairs of frames. The sequences record a single person performing a variety of gestures with different backgrounds and lighting conditions.

Rendered Hand Pose (RHP). The RHP dataset (Zimmermann and Brox, 2017) provides 41258 and 2728 images for training and testing, respectively. All images are generated synthetically using a blending software and come with accurate 2D and 3D annotations of 21 keypoints. The dataset contains 20 different characters performing 39 actions with different lighting conditions, backgrounds, and camera viewpoints.

MPII+NZSL. The MPII+NZSL dataset (Simon et al., 2017) provides 2800 2D hand pose annotations for in-the-wild images. The images are taken from YouTube videos of people performing daily life activities and New Zealand Sign Language exercises. The dataset is split into 2000 and 800 images for training and testing, respectively. In addition, (Simon et al., 2017) also provides additional training data that contains 14261 synthetic images and 14817 real images. The annotations for real images are generated automatically using multi-view bootstrapping. We refer to these images as MVBS in the rest of this chapter.

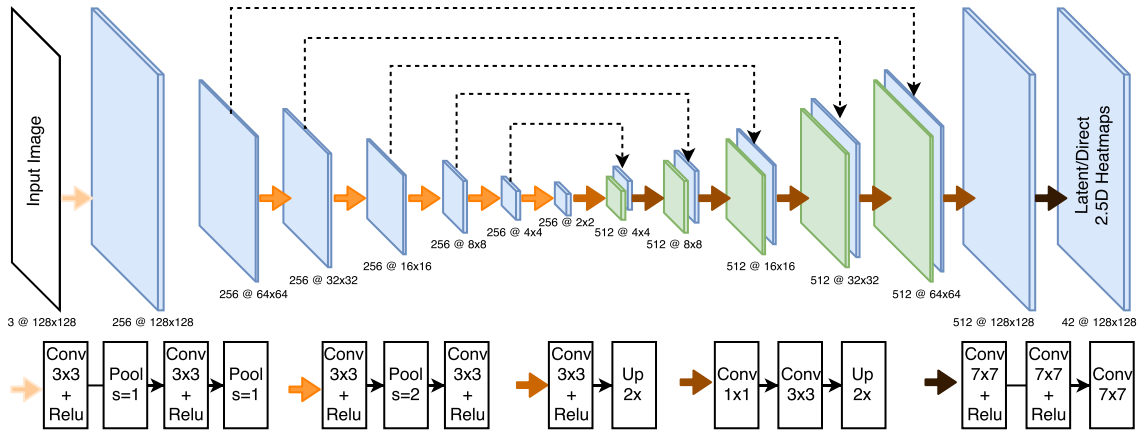


Figure 9.3: Backbone network used for 2.5D heatmap regression.

9.4.1. Evaluation Metrics

For our evaluation on the D+O, ED, SHP, and RHP datesets, we use average End-Point-Error (EPE) and the Area Under the Curve (AUC) on the Percentage of Correct Keypoints (PCK). We report the performance for both 2D and 3D hand pose where the performance metrics are computed in pixels and millimeters (mm), respectively. We use the publicly available implementation of evaluation metrics from (Zimmermann and Brox, 2017). For the D+O and ED datasets, we follow the evaluation protocol proposed by Mueller et al. (2018), which requires estimating the absolute 3D pose with global scale. For SHP and RHP, we follow the protocol proposed by Zimmermann and Brox (2017), where the root keypoints of the ground-truth and estimated poses are aligned before calculating the metrics. For the MPII+NZSL dataset, we follow Simon et al. (2017) and report head-normalized PCK (PCKh) in our evaluation.

9.4.2. Implementation Details

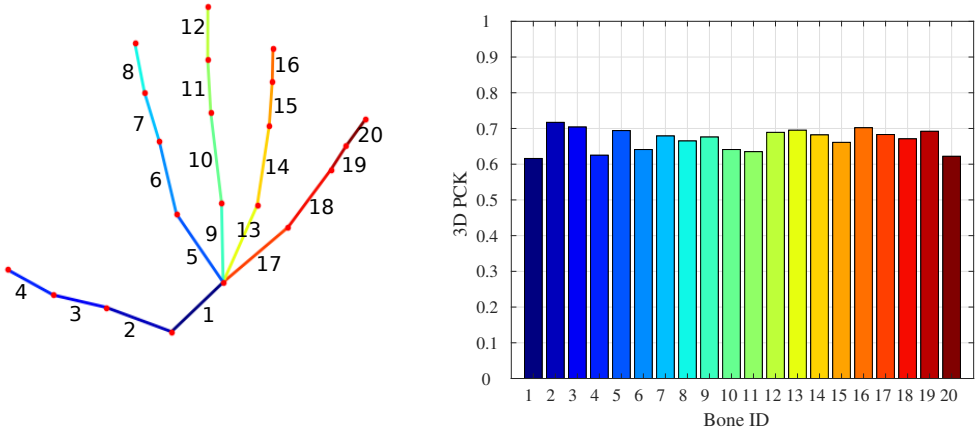
9.4.2.1. Holistic 2.5D Regression

We follow Sun et al. (2017b) and use a ResNet-50 (He et al., 2016) model for holistic regression. As in (Sun et al., 2017b), we mean normalize the poses before training and use L_1 norm as the loss function. The input to the network is a 224×224 image. We use $\alpha = 1$ since the poses are normalized and the range of $\mathcal{L}_{xy}$ and $\mathcal{L}_{\hat{z}}$ is similar. The initial learning rate is set to 0.03.

9.4.2.2. 2.5D Heatmap Regression

For 2.5D heatmap regression we use an Encoder-Decoder network architecture with skip connections (Ronneberger et al., 2015; Newell et al., 2016) and fixed number of channels (256) in each convolutional layer. The detailed network architecture can be seen in Figure 9.3. The input to our model is a 128×128 image, which produces 2.5D heatmaps as output with the same resolution as the input image.

For direct heatmap regression, we use $\sigma = 5$ to create the target heatmaps for training. We follow (Wei et al., 2016; Newell et al., 2016) and use L_2 norm as the loss function. The initial



(a) (b)

Figure 9.4: (a) Skeleton of the hand used in this work with bone ids. (b) Impact of the bone used for normalization in (9.2) and for reconstruction of 3D pose from 2.5D (Section 9.2.2).

learning rate is set to 0.0001.

For latent 2.5D regression, the $\mathcal{L}_{xy}$ is calculated on 2D pixel coordinates and $\mathcal{L}_{\hat{z}^r}$ is computed on the scale normalized root-relative depths. Therefore, a balancing factor is required. We empirically chose $\alpha = 20$ such that both losses have a similar magnitude. In our experiments we also tried with $\alpha = 1$ and the performance dropped insignificantly by less than 1%. We use L_1 norm as the loss function with a learning rate of 0.001. The overview of the two-stage model for 2.5D heatmap regression can be seen in Figure 9.5

9.4.2.3. Common details

We train the models only for the right hand, and during inference, flip the left hand images before passing them to the network. All models are trained from scratch with a batch size of 32 for 70 epochs. During training, we crop the bounding box such that the hand is 70% of the image. We perform data augmentation by rotation $(0, 9.0 \times 10^{1\circ})$, translation ($\pm$20 pixel), scale (0.7,1.1), and color transformations. In addition, in order to make the models robust against object occlusions, we follow Mueller et al. (2017) and randomly add textured objects (ovals and cubes) to the training samples. We decay the learning rates for all models by a factor of 10 after every 30 epochs, and use SGD with $\text{momentum} = 0.9$. During mixed training, the training images with 2D-only or 3D annotations are sampled with equal probability.

For all the video datasets, *i.e.*, D+O, ED, SHP we use the YOLO detector (Redmon and Farhadi, 2017) to detect the hand in the first frame of the video, and generate the bounding box in the subsequent frames using the estimated pose of the previous frame. We trained the hand detector using the training sets of all aforementioned datasets.

Method	2D Pose Estimation AUC ↑	2D Pose Estimation EPE (px) median ↓	2D Pose Estimation EPE (px) mean ↓	3D Pose Estimation AUC ↑	3D Pose Estimation EPE (mm) median ↓	3D Pose Estimation EPE (mm) mean ↓
Comparison with baselines						
Holistic 2.5D reg.	0.41	17.34	22.21	0.54	42.76	47.80
Direct 2.5D heatmap reg.	0.57	10.33	21.63	0.55	36.97	52.33
Latent 2.5D heatmap reg. (Ours)	0.59	9.91	16.67	0.57	39.62	45.54
Impact of training data						
Latent 2.5D heatmap regression trained with						
SHP + RHP	0.59	9.91	16.67	0.57	39.62	45.54
+ MPII + NZSL	0.67	9.07	10.65	0.68	28.11	32.78
+ MVBS	0.68	8.84	10.45	0.68	27.27	32.75
Comparisons with the baselines with additional training data trained with SHP+RHP (3D pose) and MPII+NZSL+MVBS (2d pose) datasets						
Holistic reg.	0.53	12.98	16.17	0.66	31.71	34.86
Direct heatmap reg.	0.65	9.60	12.06	0.68	25.92	35.56
Latent heatmap reg.	0.68	8.84	10.45	0.68	27.27	32.75
Performance after removing labeling discrepancy						
Holistic regression	0.59	10.66	14.10	0.67	30.69	33.80
Direct heatmap reg.	0.72	7.05	9.66	0.68	25.37	34.88
Latent heatmap reg.	0.76	5.95	7.97	0.69	26.56	31.86
Latent heatmap reg. - fast	0.71	6.44	10.67	0.68	28.08	33.35

Table 9.1: Ablation studies. The arrows specify whether a higher or lower value for each metric is better. The first block compares the proposed approach of latent 2.5D heatmap regression with two baseline approaches. The second block shows the impact of different training data and the last block shows the impact due to differences in the annotations.

9.4.3. Ablation Studies

We evaluate the proposed method under different settings to better understand the impact of different design choices. We chose the D+O dataset for all ablation studies, mainly because it does not have any training data. Thus, it allows us to evaluate the generalizability of the proposed method. Finally, since the palm (root) joint is not annotated, it makes it compulsory to estimate the absolute 3D pose in contrast to the commonly used root-relative 3D pose. We use (9.8) to estimate the global scale of each 3D pose using the mean bone lengths from the SHP dataset.

The ablative studies are summarized in Table 9.1. We first examine the impact of different choices of CNN architectures for 2.5D pose regression. For holistic 2.5D pose regression, we use the commonly adopted (Sun et al., 2017b) ResNet-50 (He et al., 2016) model. Using a holistic regression approach results in an AUC of 0.41 and 0.54 for 2D and 3D pose, respectively. Directly regressing the 2.5D heatmaps significantly improves the performance of 2D pose estimation (0.41 vs. 0.57), while also raising the 3D pose estimation accuracy from 0.54 AUC to 0.55. Using latent heatmap regression improves the performance even further to 0.59 AUC and 0.57 AUC for 2D and 3D pose estimation, respectively. While the holistic regression approach achieves a competitive accuracy for 3D pose estimation, the accuracy for 2D pose estimation is inferior to the heatmap regression due to

its limited spatial output resolution.

We also evaluate the impact of training the network in a multi-task setup. For this, we train the model with additional training data from (Simon et al., 2017) which provides annotations for 2D keypoints only. First, we only use the 2000 manually annotated real images from the training set of the MPII+NZSL dataset. Using additional 2D pose annotations significantly improves the performance. Adding additional $15,000$ annotations of real images from the MVBS dataset (Simon et al., 2017) improves the performance only slightly. Hence, only 2000 real images are sufficient to generalize the model trained on synthetic data to a realistic scenario. We evaluate the impact of additional training data on all CNN architectures for 2.5D regression. We can see that the performance improves for all architectures, but importantly, the ordering in terms of performance stays the same.

The annotations of the fingertips in the D+O dataset are slightly different than in the other datasets. In the D+O dataset, the fingertips are annotated at the middle of the tips whereas other datasets annotate it at the edge of the nails. To remove this discrepancy, we shorten the last bone of the fingertip by 0.9. Fixing the annotation differences results in further improvements, revealing the true performance of all approaches. Note that the ordering of the methods still stays the same, and our proposed approach for latent heatmap regression outperforms others options.

In Figure 9.4b, we evaluate the impact of pair of keypoints (bones) selected for 3D pose normalization in (9.2). The skeleton of the hand used in this work can be seen in Figure 9.4a. For this, we trained a separate CNN model while using a specific pair of keypoints for normalization. We used latent heatmap regression for this experiment. We can see that the performance remains consistent (≈ 0.69) for most of the bones.

We also evaluate the runtime of the used models on an NVIDIA TitanX Pascal GPU. While the holistic 2.5D regression model runs at 145 FPS, direct and latent 2.5D heatmap regression networks run at 20 FPS. We trained a smaller model with 128 feature maps (base layers) and replaced 7x7 convolutions in the last 3 layers with 1x1, 3x3 and 1x1 convolutions. The simplifications resulted in 150 FPS and parameter reduction by 3.8x while remaining competitive to direct heatmap regression with the full model. This model is marked with label "fast" in Table 9.1.

We also evaluate the impact of using multiple stages in the network, where each stage produces latent 2.5D heatmaps as output. While the first stage only uses the features extracted from the input image using the initial block of convolutional layers, each subsequent stage also utilizes the output of the preceding stage as input. This provides additional contextual information to the subsequent stages and helps in incrementally refining the predictions. Similar to (Newell et al., 2016; Wei et al., 2016) we provide local supervision to the network by enforcing the loss at the output of each stage. An overview of the two-stage network architecture can be seen in Figure 9.5. Adding one extra stage to the network increases the 3D pose estimation accuracy from AUC 0.69 to 0.71, but decreases the 2D pose estimation accuracy from AUC 0.76 to 0.74. The decrease in 2D pose estimation accuracy is most likely due to over-fitting to the training datasets. Remember that we do not use any training data from the D+O dataset. In the rest of this chapter, we always use networks with two stages unless stated otherwise.

Finally, in Table 9.2, we also compare the performance of direct and latent heatmap regression on other datasets. The performance of proposed latent heatmap regression remains consistently superior as compared to direct heatmap regression.

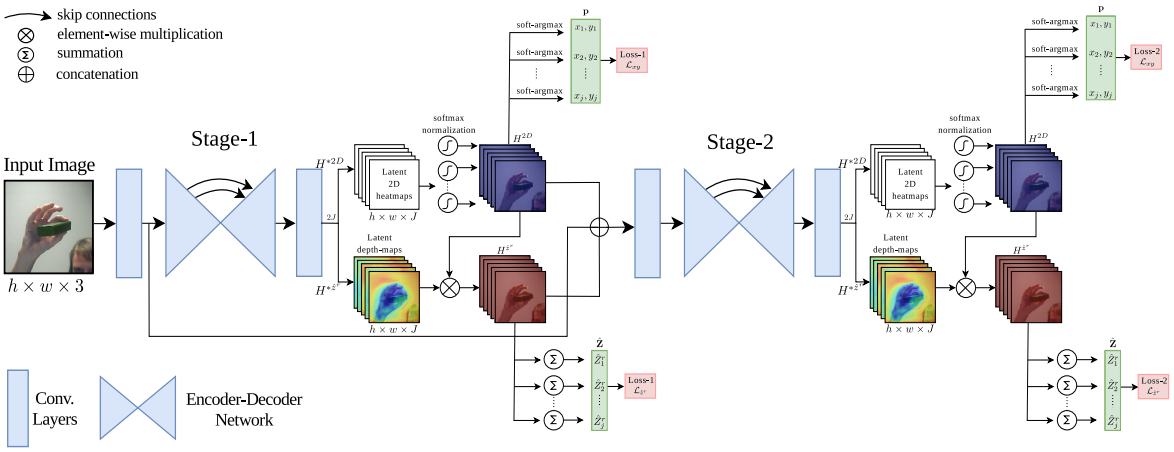


Figure 9.5: Overview of the two stage model for latent 2.5D heatmap regression.

Method	RHP		SHP		MPII+NZSL
	2D	3D	2D	3D	2D
Direct heatmap reg.	0.88	0.84	0.78	0.99	0.91
Latent heatmap reg.	0.88	**0.88**	**0.80**	0.99	**0.92**

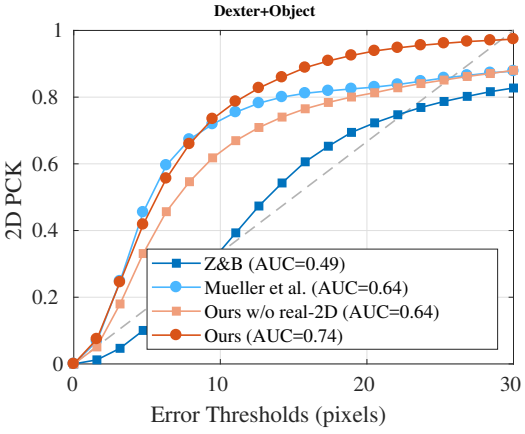
Table 9.2: Results for additional datasets measured in terms of AUC.

9.4.4. Comparison to State-of-the-Art

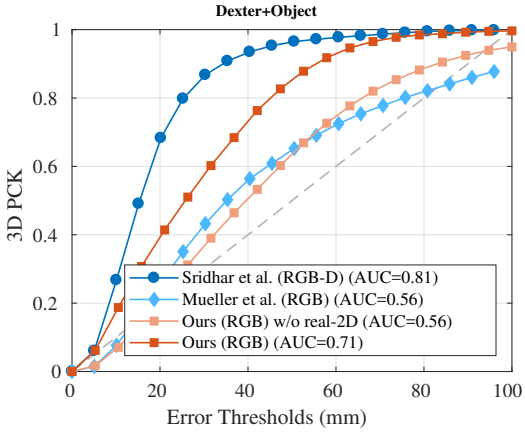
We provide a comparison of the proposed approach with state-of-the-art methods on all aforementioned datasets. Note that different approaches use different training data. We thus replicate the training setup of the corresponding approaches for a fair comparison.

Figure 9.6a and 9.6b compare the proposed approach with other methods on the D+O dataset for 2D and 3D pose estimation, respectively. In particular, we compare with the state-of-the-art method by Zimmerman and Brox (Z&B) (Zimmermann and Brox, 2017) and the contemporary work by Mueller *et al.* (Mueller et al., 2018). We use the same training data (SHP+RHP) for comparison with (Zimmermann and Brox, 2017) (AUC 0.64 vs 0.49), and only use additional data for comparison with (Mueller et al., 2018)(AUC 0.74 vs 0.64). For the 3D pose estimation accuracy (Figure 9.6b), the approach (Zimmermann and Brox, 2017) is not included since it only estimates scale normalized root-relative 3D pose. Our approach clearly outperforms current RGB state-of-the-art method by Mueller *et al.* (Mueller et al., 2018) by a large margin. The approach (Mueller et al., 2018) utilizes the video information to perform temporal smoothening and also performs subject specific adaptation under the assumption that the users hold their hand parallel to the camera image plane. In contrast, we only perform frame-wise predictions without temporal filtering or user assumptions. Additionally, we report the results of the depth based approach by Sridhar *et al.* (Sridhar et al., 2016), which are obtained from (Mueller et al., 2018). While the RGB-D sensor based approach (Sridhar et al., 2016) still works better, our approach takes a giant leap forward as compared to the existing RGB based approaches.

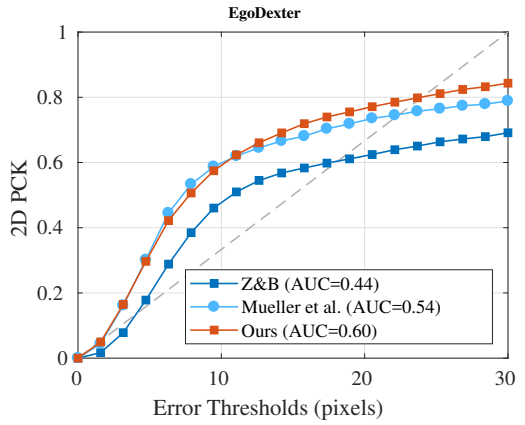
Figure 9.6e compares the proposed method with existing approaches on the SHP dataset. We use the same training data (SHP+RHP) as in (Zimmermann and Brox, 2017) and outperform all existing



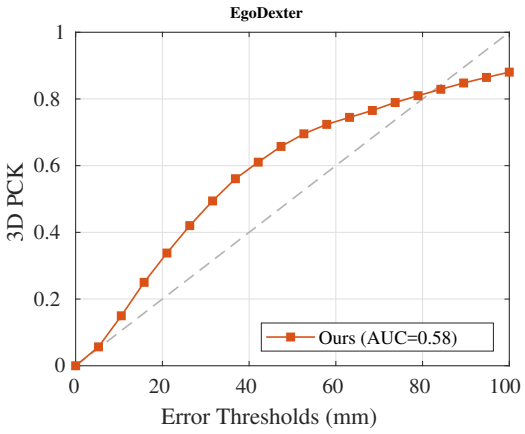
(a) 2D PCK on D+O dataset.



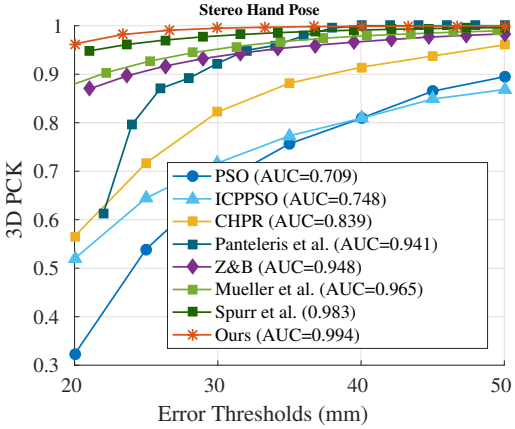
(b) 3D PCK on D+O dataset.



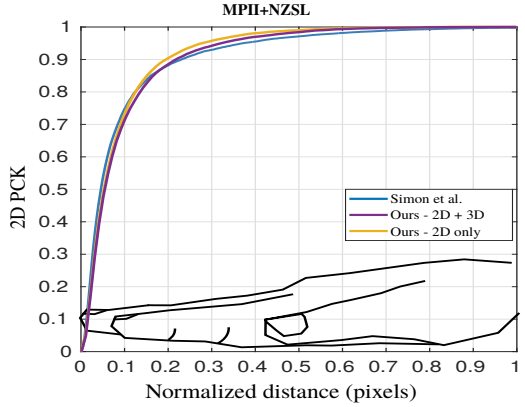
(c) 2D PCK on ED dataset.



(d) 3D PCK on ED dataset.



(e) 3D PCK on SHP dataset.



(f) 2D PCKh on MPII+NZSL.

Figure 9.6: Comparison with the state-of-the-art on the DO, ED, SHP and MPII+NZSL datasets.

Method	2D Pose Estimation			3D Pose Estimation		
	AUC ↑	EPE (mm) median ↓	EPE (mm) mean ↓	AUC ↑	EPE (mm) median ↓	EPE (mm) mean ↓
(Zimmermann and Brox, 2017)	0.72	5.00	9.14	-	18.8*	-
(Spurr et al., 2018)	-	-	-	0.85	19.73	-
Ours	0.89	2.20	3.57	0.91	13.82	15.77
Ours w. GT $\hat{Z}_{root}$ and $\hat{s}$	0.89	2.20	3.57	0.94	11.33	13.41

Table 9.3: Comparison with the state-of-the-art on the RHP dataset. *uses noisy ground-truth 2D poses for 3D pose estimation.

methods despite the already saturated accuracy on the dataset and the additional training data and temporal information used in (Mueller et al., 2018).

Figure 9.6c compares the 2D pose estimation accuracy on the EgoDexter dataset. While we outperform all existing methods for 2D pose estimation, none of the existing approaches report their performance for 3D pose estimation on this dataset. We, however, also report our performance in Figure 9.6d.

The results on the RHP dataset are reported in Table 9.3. The proposed method significantly outperforms state-of-the-art approaches (Zimmermann and Brox, 2017; Spurr et al., 2018). Since the dataset provides 3D pose annotations for complete hand skeleton, we also report the performance of the proposed method when the ground-truth depth of the root joint and the global scale of the hand is known (w. GT $\hat{Z}_{root}$ and $\hat{s}$). We can see that our approach of 3D pose reconstruction and scale recovery is very close to the ground-truth.

Finally, for completeness, in Figure 9.6f we compare our approach with (Simon et al., 2017) which is a state-of-the-art approach for 2D pose estimation. The evaluation is performed on the test set of the MPII+NZSL dataset. We follow (Simon et al., 2017) and use the provided center location of the hand and the size of the head of the person to obtain the hand bounding box. We define a square bounding box with height and width equal to $0.7 \times head\text{-}length$. We report two variants of our method; 1) the model trained for both 2D and 3D pose estimation using the datasets for both tasks, and 2) a model trained for only 2D pose estimation using the same training data as in (Simon et al., 2017). In both cases we use the models trained with 2-stages. Our approach performs similar or better than (Simon et al., 2017), even though we use a smaller backbone network as compared to the 6-stage Convolutional Pose Machines (CPM) network (Wei et al., 2016) used in (Simon et al., 2017). The CPM model with 6-stages has $51M$ parameters, while our 1 and 2-stage models have only $17M$ and $35M$ parameters, respectively. Additionally, our approach also infers the 3D hand pose.

Some qualitative results for 3D hand pose estimation for in-the-wild images can be seen in Figure 9.7 and some video results can also be seen at `https://youtu.be/4Q3ByHZ8tNc`.

9.5. SUMMARY

We have presented a method for 3D hand pose estimation from a single RGB image. We demonstrated that the absolute 3D hand pose can be reconstructed efficiently from a single image up to a scaling factor. We presented a novel 2.5D pose representation which can be recovered easily from

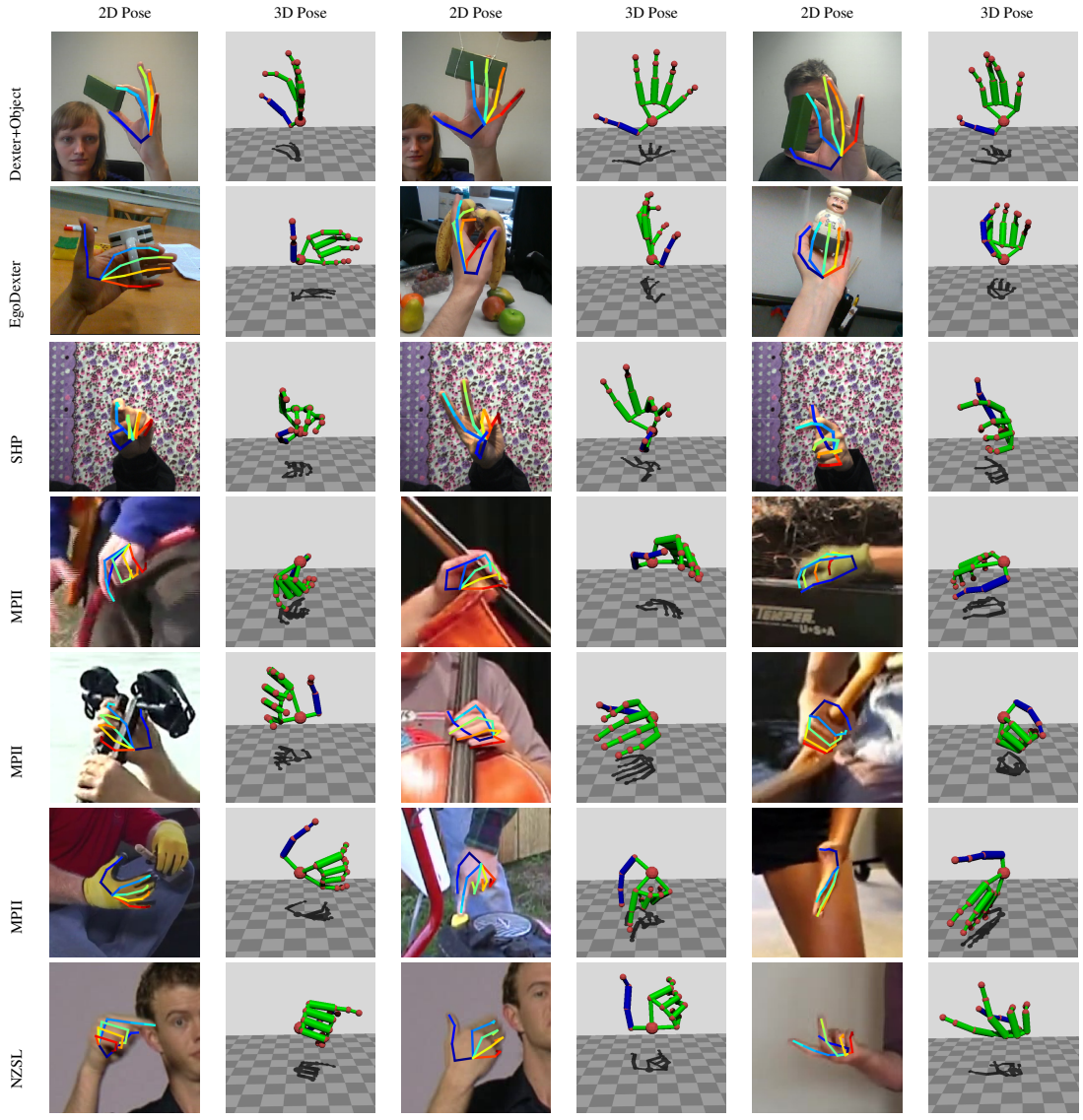


Figure 9.7: Qualitative Results. The proposed approach can handle severe occlusions, complex hand articulations, and unconstrained images taken from the wild.

RGB images since it is invariant to absolute depth and scale ambiguities. It can be represented as 2.5D heatmaps, therefore, allows keypoint localization with sub-pixel accuracy. We also proposed a CNN architecture to learn 2.5D heatmaps in a latent way using a differentiable loss function. Finally, we proposed an approach to reconstruct the 3D hand pose from 2.5D pose representation. The proposed approach demonstrated state-of-the-art results on five challenging datasets with severe occlusions, object interactions and images taken from the wild.

CHAPTER 10

Conclusion

Contents

10.1. OVERVIEW

The goal of this thesis was to advance the field of 2D and 3D articulated human pose estimation in unconstrained images and videos.

2D human body pose estimation has seen remarkable progress over the last few years. The main success stemmed from stronger appearance modeling using deep learning methods and availability of large-scale datasets. However, most of the earlier methods and available datasets focused only on the pose estimation of single persons and ignored people in the crowd. Further, the pose estimation and tracking of multiple people in videos, albeit being essential for most of the practical applications, did not receive much attention in the literature. This thesis addressed these more challenging cases of multi-person 2D human body pose estimation. We proposed methods that can work in entirely unconstrained scenarios, established a large-scale benchmark, developed comprehensive evaluation metrics, and conducted extensive experiments to analyze the developed baselines and approaches.

In case of 3D human body pose estimation from single images, successful approaches use regression-based methods and rely on training images with annotated 3D poses. The existing datasets with 3D pose annotations are, however, either recorded in controlled indoor settings, or consist of synthetically generated images. Hence, the methods trained on these datasets cannot generalize well in unconstrained scenarios. We instead proposed an approach that does not require any training data with 3D pose annotations. We built on the success in 2D pose estimation and combined it with a robust method for 3D pose retrieval. This allowed us to circumvent the requirement of training data and to perform 3D pose estimation in the images taken from the wild. We demonstrated the effectiveness of the proposed method with extensive experiments.

In the last chapter we proposed an approach for 3D pose estimation via a novel 2.5D representation. Our 2.5D pose representation can be estimated reliably from RGB images and allows to reconstruct the scale normalized absolute 3D pose. However, since very few works in the literature

have addressed the problem of hand pose estimation from RGB images, we shifted our attention to hands in this chapter. Our proposed method achieved state-of-the-art results on challenging benchmark datasets, and works under sever occlusion, complex articulations and unconstrained images.

Additionally, we contributed to the research community by releasing the source code of the methods proposed in this thesis. We also publicly released the annotated datasets. Further, we hosted an international challenge and workshop[1] to attract the attention of the research community to the challenging problems addressed in this thesis. Finally, for a fair evaluation of developed methods for multi-person pose tracking, we developed an online server to evaluate the results on a held-out testset. We believe that the outcomes of this thesis in term of research, open-source codes, data, and workshops will help to further progress on these challenging topics in computer vision.

10.2. CONTRIBUTIONS AND DISCUSSION

The work presented in this thesis contributed towards three main directions, as presented below.

10.2.1. 2D Human Body Pose Estimation

In Chapter 4, we contributed an efficient approach for multi-person 2D human body pose estimation. We argued that solving the problem of joint-to-person association globally for all persons, as done in the other approaches (Pishchulin et al., 2016), is expensive and avoidable. We showed that the problem can be formulated by a set of independent local joint-to-person association problems. Further, we showed that the non-maxima suppression and joint labeling can be performed directly using a CNN. Our proposed modifications resulted in local optimization problems that can be solved efficiently as compared to global modeling, while still being robust to sever occlusions and truncations. We evaluated our approach on the benchmark dataset where we improved the accuracy of (Pishchulin et al., 2016) while also reducing the runtime by a factor between 6,000 and 19,000.

In Chapter 5, we formally introduced the problem of joint multi-person pose estimation and tracking. We presented a challenging and unconstrained annotated dataset of videos with 16K pose annotations. We selected the videos such that they contain diverse, complex and realistic scenarios. We contributed a comprehensive evaluation protocol to evaluate multi-person pose estimation and tracking jointly. To this end, we followed the best practices followed in both pose estimation (Pishchulin et al., 2016) and multi-target tracking (Milan et al., 2016). We devised the evaluation protocol such that the developed algorithms cannot make any assumption about the number, size, or location of the persons. Following the protocol, we also presented an approach that can perform pose estimation and tracking in completely unconstrained videos. To this end, we built on the approaches in Chapter 4 and (Insafutdinov et al., 2016), and formulated the problem as a spatio-temporal graph whose feasible solution directly provides the poses and track-ids of all persons visible in the video. We showed that the graph can be efficiently optimized using integer linear programming. We compared our approach with other baselines and demonstrated that our approach performs better, and is robust to sever occlusions and truncation. We released our dataset and source code for public usage.

In Chapter 6, we further extended our dataset to a large-scale benchmark. We enlarged it from 60 videos to 550 videos. Compared to 16K poses in the previous version, the newer dataset contains

[1] The 2nd PoseTrack workshop will be held in conjunction with ECCV'18 in Munich, Germany. This time the workshop will also host challenges for 3D and dense pose estimation.

more than 150K pose annotations. We selected the videos such that they contain a large amount of body pose, appearance, and scale variation, as well as body part occlusion and truncation. Further, we ensured that the videos contain severe body motion, *i.e.*, people occlude each other, re-appear after complete occlusion, vary in scale across the video, and also significantly change their body pose. We also ensured that the selected videos contain cases where the number of visible persons and body parts changes over time. To attract the attention of the research community towards this challenging problem, we organized a PoseTrack challenge and workshop at ICCV'17 and invited the research community to submit novel solutions for multi-person pose estimation and tracking. We also developed two baseline methods. For this, we modified our approach from Chapter 5 to incorporate stronger pose estimation model (Cao et al., 2017) and developed an additional baseline method based on (Insafutdinov et al., 2017). We evaluated the approaches submitted to PoseTrack challenge and our baselines on the new dataset and conducted extensive performance analysis. We highlighted the strengths and weaknesses of the state-of-the-art approaches and highlighted the most promising future research directions. To prevent over-fitting and to ensure that all methods will be evaluated using the same ground-truth and evaluation scripts, we kept the test data with-held and developed an online evaluation server. To-date, ~400 researchers from ~350 institutes[2] have registered at our online server, and our dataset got downloaded for more than 4,000 times.

In Chapter 7, we contributed an approach that refines body pose trajectories by exploiting high-level information about human activities. First, we showed that the activity of the person can be estimated reliably by using its body pose trajectory. Then, we demonstrated that the obtained information about the activity can be subsequently used to incorporate higher-order part dependencies for the refinement of initial poses. For this, we proposed an action-conditioned pictorial structure model which starts with a uniform prior and updates this prior based on the probabilities from a pose-based action classifier. We also showed that learning the right amount of appearance sharing among action classes improves the pose estimation accuracy. We also incorporated Convolutional Channel Features (Yang et al., 2015) from a pre-trained VGG network (Simonyan and Zisserman, 2014), and showed that using CCF with the random forest based part regressors significantly improves the performance. We benchmarked our approach on two datasets and demonstrated superior performance as compared to our direct competitors (Dantone et al., 2014; Yang and Ramanan, 2013; Cherian et al., 2014; Nie et al., 2015; Chen and Yuille, 2014). In contrast to other chapters, in this work we chose the pictorial structure model as a baseline. Our experiments on comparing stronger and weaker baselines showed that action information is helpful for pose estimation regardless of the strength of the underlying baseline model. We expect that integrating action information in the CNN based models would lead to further performance gains. However, investigating the approach to effectively incorporate this information into the commonly adopted fully-convolutional neural networks remains an open question, and we plan to explore this direction in the future work.

10.2.2. 3D Human Body Pose Estimation

In Chapter 8, we contributed a dual-source approach for 3D pose estimation. We showed that the problem of 3D pose estimation can be decomposed into two sub-problems of 2D pose estimation and 3D pose retrieval. We showed that such a decomposition does not require any training data with 3D pose annotations. Instead it can rely on two independent sources of training data both of which

[2]Number of unique domains

are available abundantly. For 2D pose estimation, we used a dataset with 2D pose annotation to train a CNN based pose estimation model. For 3D pose retrieval, we converted a motion capture data to a normalized 2D pose space. We demonstrated that during inference the nearest 3D poses can be retrieved efficiently by searching for the nearest neighbors in the normalized 2D pose space using kd-tree. Given the nearest 3D poses, we proposed a novel objective function to jointly estimate a mapping from the 3D pose space to the image and reconstruct the 3D pose. We demonstrated the effectiveness of our approach with extensive experiments. We showed that our approach is not bound to any specific dataset and it can generalize across different scenarios. In contrast to existing methods, our approach is shown to work well even when the training data for both sub-problems are from very different sources or have a different skeleton structure. To ensure reproducibility of our work, we also released our source code for public usage.

10.2.3. Hand Pose Estimation

Finally, in Chapter 9, we presented an approach for hand pose estimation from RGB images. We contributed a novel 2.5D pose representation along with an exact approach for 3D pose reconstruction, and also a novel CNN architecture. We showed that the problem of 3D hand pose estimation can be decomposed into two subproblems of 2.5D pose regression and reconstruction of 3D pose from 2.5D. We showed that the scale normalized absolute 3D hand pose can be recovered from 2.5D pose representation without ambiguities. Further, we demonstrated that our 2.5D pose representation allows training the network in a multi-task setup, and showed the impact of utilizing multiple sources of training data. We also showed that 2.5D pose can be estimated precisely by regressing 2.5D heatmaps in a latent way. We validated the merits of our approach on five benchmark datasets for 2D and 3D hand pose estimation where it achieved state-of-the-art results. Finally, we demonstrated with qualitative results that our approach can handle severe occlusions, complex hand articulations, and unconstrained images taken from the wild.

Although we evaluated the approaches proposed in Chapter 8 and 9 on two different problems, both approaches can be interchangeably used for body or hand pose estimation. Given a 2D hand pose estimation model, hand poses from the BigHand2.2M dataset (Yuan et al., 2017), for example, can be used as a MoCap data in Chapter 8 for 3D pose retrieval and reconstruction. Similarly, the MPII Pose (Andriluka et al., 2014) and Human3.6M datasets can be combined to train the network for 2.5D pose regression in Chapter 9, where the size of the head bounding-box can be used to obtain the normalized 2.5D pose representation and for 3D pose reconstruction.

Interestingly, both methods also provide different benefits in different scenarios. For example, in Chapter 8 we reconstruct 3D poses only from 2D pose information. This approach is beneficial when no training images with 3D pose annotations are available. Whereas, in Chapter 9, we demonstrated that the networks can be trained in a multi-task setup to efficiently combine unconstrained images with 2D pose annotations and constrained images with 3D pose annotations. In particular, we showed that only 2000 real images of hands with 2D pose annotations are sufficient to generalize the model trained on synthetic data to a realistic scenario. This is beneficial when at least some kind of training images with 3D pose annotations are available regardless of the environment in which they were captured. Since the approach in Chapter 9 can better handle the re-projection ambiguities, it is a preferred approach whenever the required training data is available.

10.3. FUTURE WORK

Handling Complex Articulation

In Chapter 6, we highlighted that the current methods for body pose estimation struggle to accurately estimate the poses with complex articulations (see Figure 6.7). This is mainly due to the long tail distribution of the training samples. Current methods treat all images in the training data equally and performing hard-negative mining for CNN based models is impractical due to their long training time. To this end, exploring methods for online hard negative mining to detect and assign higher weights to the difficult poses during training is an interesting future work.

Task-Specific Matching Algorithm for Multi-Person Pose Tracking

The choice of the similarity metrics used for multi-person pose tracking plays a crucial role in the final performance. In Chapter 5 and 6, we used dense correspondences based on optical flow information (Weinzaepfel et al., 2013) or pose based similarity to measure the similarity between two poses. More recent methods for pose tracking also rely on non-parametric metrics such as PCKh (Girdhar et al., 2018) or OKS (Xiao et al., 2018) between a pair of poses, IoU between the person bounding boxes (Girdhar et al., 2018), the similarity between the image features (Girdhar et al., 2018) or the optical flow information (Insafutdinov et al., 2017). The location-based metrics such as PCKh, OKS or IoU assume that the poses change smoothly over time, and therefore, struggle in case of a large camera or body pose motion and scale variations due to camera zoom. On the other hand, appearance-based similarity metrics or optical flow information cannot handle large appearance variations due to person occlusions or truncation, motion blur, etc. In Chapter 5, we tried to tackle these challenges by enforcing long-range temporal coherence by formulating the problem using a complex spatio-temporal graph which, however, resulted in a very high inference time, and can therefore be infeasible for applications that require online pose tracking. Further, all of these metrics are agnostic to the underlying task, therefore, can be sub-optimal.

The main reason behind using task-agnostic similarity metrics was the unavailability of enough training data. However, with our newly proposed PoseTrack dataset in Chapter 6, learning task-specific similarity metrics should be possible. We explored this direction in a preliminary work (Doering et al., 2018), where we propose a task-specific novel representation for person association over time. We refer to this representation as Temporal Flow Fields (TFF). TFF represent the movement of each body part between two consecutive frames using a set of 2D vectors encoded in an image. Our TFF representation is inspired by the Part Affinity Fields representation (Cao et al., 2017) that measures the spatial association between different body parts and is learned by a CNN. We integrate TFF in an online multi-person tracking approach and demonstrate that a greedy matching approach is sufficient to obtain good tracking results. Using the same pose estimation model, TFF with a very simple greedy matching approach for tracking can already outperform our approaches in Chapter 6 as shown in Tab. 10.1.

This demonstrates that having a task-specific similarity metric alone can lead to a significant performance gain. We envision that combining TFF with a more sophisticated spatio-temporal graph for person association, as used in Chapter 5&6 would lead to further improvements. In the future, we plan to explore this direction further and hope to develop stronger representations for similarity metrics. We refer the readers to (Doering et al., 2018) for more details on TFF.

Method	MOTA	mAP
PoseTrack (Chapter 6)	48.4	59.4
ArtTrack (Chapter 6)	48.1	59.4
TFF (Doering et al., 2018)	53.1	63.3

Table 10.1: Comparison of TFF with PoseTrack (Chapter 6) and ArtTrack (Chapter 6) based baselines. Utilizing TFF without using a sophisticated spatio-temporal graph already leads to significant improvement which shows the importance of task-specific similarity metrics.

Handling Large Motion

As shown in Chapter 6, another limitation of the current methods for multi-person pose tracking is their lack of ability to handle the large camera or person movements. The task-specific similarity metrics learned using a convolutional neural network as described above can only partially address this problem. This is because the convolutional networks have an inherent limitation in that they only utilize information from a local neighborhood. Therefore, in case of large person or camera movements, it is possible that the receptive field of the convolutional kernels does not cover the regions of interest between two frames. To this end, recent methods in object detection employ deformable convolutional kernels that deform their structure and adapt the receptive field based on the observed image information (Dai et al., 2017). Exploiting deformable convolutional kernels that adapt their sampling locations and receptive field based on the motion information is an interesting future work to handle videos with large motion.

End-to-End Multi-Person Pose Estimation and Tracking

In Chapter 6, we highlighted that most of the existing methods for pose estimation and tracking build on the combination of disjoint components and none of the approaches is end-to-end in the sense that it can infer articulated people tracks from video directly. The main reason is that all existing methods for multi-person pose estimation contain at least one non-differentiable component before generating the poses. This restricts these networks from performing any additional reasoning (*i.e.*, activity recognition, motion anticipation or tracking) in an end-to-end way. In Chapter 9, we presented a network architecture that converts the body part heatmaps to pose coordinates in a differentiable way. Exploring similar network architectures that can generate body poses of a variable number of people and associate them overtime in a fully-differentiable manner is an interesting future direction.

Self-Supervised Matching for Multi-Person Pose Tracking

In this thesis we contributed a large-scale dataset for multi-person pose estimation and tracking which consists of more than 150K pose annotations for more than 500 videos. The dataset has stimulated a huge interest from the community towards this challenging problem. However, annotating this dataset was a laborious task and even with effective annotation tool and more than thirty annotators, it took us several months to complete the annotations. Despite such a huge effort, the resulting 500 annotated videos are still insufficient to effectively train fully-supervised deep neural networks. This, particularly, is the reason that none of the state-of-the-art approaches for multi-person pose tracking

perform any learning on the video sequences provided in our dataset (see. Chapter 6). The main reason is that the videos contain redundant appearance information, whereas training of networks that generalize to other datasets require large amounts of appearance variation in the training data. To this end, an interesting future direction is to devise approaches to train the networks in a self-supervised manner. Several recent works have demonstrated that the networks addressing the problems such as optical flow (Ranjan et al., 2018; Meister et al., 2017; Yin and Shi, 2018) and even object tracking (Vondrick et al., 2018) can be learned in a self-supervised way. Since multi-person pose tracking shares many characteristics with these problems, similar approaches can be explored in this direction.

Multi-Person Pose Tracking via Segmentation of Part Trajectories

In this thesis, we always relied on the image based matching algorithms to track the persons. However, the studies on the visual perception of biological human motion have shown that the movement of a sparse set of keypoints without any color information is sufficient for humans to track and distinguish different persons (Fischler and Elschlager, 1973). An interesting question here is whether we can transform similar capabilities into computational methods. Inspired by these studies on the human visual system, an interesting direction is to treat the tracking problem as the segmentation of body part trajectories into unique individuals. One great benefit that such an approach will bring is that it does not require any annotated images. But instead, it only needs a vast range of body pose trajectories which can be simulated or generated easily from existing annotated or MoCap datasets.

Action Priors for Human Pose Estimation

In Chapter 7, we presented an action condition pictorial structure model that incorporates information about human activities to improve body pose estimation. We expect that further performance gain can be achieved by incorporating action information in stronger CNN based pose estimation models. In the following, we provide some interesting future works in this direction.

The first prerequisite to explore this direction is the availability of a large dataset with action and pose annotations. To this end, the PoseTrack dataset proposed in Chapter 6 can be extended by annotating activity labels. The videos in the PoseTrack dataset were sampled from YouTube following a two-level hierarchy of human activities (Ainsworth et al., 2011; Andriluka et al., 2014). Hence, the activity information already exists to some extent in the dataset, and some additional effort is required to organize and extend the existing video-level labels to person-level. It can also be combined with JHMDB and Penn-Action datasets to increase the number of videos. Further, there exist a large number of datasets for activity recognition, and some of them also provide person bounding boxes (*e.g.*, AVA dataset Gu et al. (2018)). These datasets can also be used as additional weak supervision. In this direction, an interesting future work is to combine the existing sources into a large-scale benchmark that provides videos for a set of well-defined action classes with different level of supervision *e.g.*, pose-only, activity-only, pose+activity, bounding+activty, etc. Such training data can be used, for example, to train the convolutional networks in a multi-task setup. Specifically, the videos with pose annotations can be used to train the pose estimation model, while the videos with activity-only annotations can be used to regularize the pose estimation model in a way that the generated poses are classified to the correct action class.

Another interesting direction is to develop methods for pose-based action recognition that can handle poses with different number of visible body parts. Existing methods for pose-based action

recognition such as used in Chapter 7 assume that full-body pose is available, which obviously is not the case in unconstrained environments. To this end, Choutas et al. (2018) recently proposed an interesting image-like representation of body pose motion which they referred to as "PoTion". PoTion can handle poses with different number of visible joints and is designed to be integrated easily with convolutional networks for activity recognition. However, it cannot handle videos with multiple interacting people. Exploring similar pose motion representation which can handle multiple people under occlusions and truncations is another interesting future work.

Another interesting direction is to investigate ways to integrate action information in CNN based pose estimation models. While the activity information, as mentioned above, can be used to regularize the pose estimation models, it does not explicitly condition the pose models on activity information. Another straightforward way is to train separate models for each activity class and select the best model conditioned on the action priors. This, however, is sub-optimal since the number of models would increase linearly with the number of action classes. Another simple way is to create a network architecture which contains multiple pose estimation branches each tailored for a particular action class. The output of all branches can be then aggregated based on action priors. The features extracted from a pre-trained action recognition model can also be used as input to the pose estimation model which will explicitly condition the network on the action information. While these are some preliminary ideas, exploring efficient ways to integrate action information in convolutional networks is an interesting future work.

Further, all the existing methods for pose-based action recognition assume that only a single person is visible in the video. Interesting future works would include methods that can jointly perform pose estimation and activity recognition of multiple interacting people while also exploiting the complementary nature of both tasks.

More Visual Cues for 3D Pose Retrieval

In Chapter 8, we utilized the 2D pose information to retrieve nearest 3D poses from a large MoCap dataset. However, utilizing only the 2D pose information is prone to re-projection ambiguities. Some recent works show that the information about the boolean geometric relationships between body parts (*e.g.*, left-wrist is in front of left-elbow, right-knee is behind right-hip, etc.) can be detected easily from the images, and help in 3D pose estimation (Pons-Moll et al., 2014; Pavlakos et al., 2018a). One great benefit of these geometric relations is that they are relatively easier to annotate for any unconstrained image and provide weak information about the 3D pose. Exploiting such geometric relationships during 3D pose retrieval can help resolve the re-projection ambiguities and is an interesting future work.

Joint-Angle Limits and Bone Ratios for 3D Pose Estimation

In Chapter 9, we presented an approach for 3D pose estimation via a 2.5D pose representation where the 2.5D pose is regressed from an image using a discriminatively trained convolutional network. The method in its current form has the drawback that it relies entirely on the pose estimation model to learn anatomically plausible poses from the training data. Since the model is not constrained to generate plausible poses, only one wrong prediction in the 2.5D coordinates can lead to an anatomically implausible pose during 3D reconstruction. To this end, an interesting future work is to explicitly exploit the available knowledge about the joint-angle limits and bone ratios of the human body to

constrain the pose estimation model. Since the joint-angle limits or bone-ratios can only be estimated from the reconstructed 3D pose, this would also require to integrate the 3D reconstruction module in the convolutional network in a differentiable way.

Self-Supervised 3D Pose Estimation

Many datasets for 3D human pose estimation (Ionescu et al., 2014b; Sigal et al., 2010) provide images from multiple views captured using calibrated cameras. Recent works for 3D keypoint localization of objects exploit such data and learn to localize the 3D keypoints in a self-supervised way by exploiting multiple domain-specific geometric cues (Suwajanakorn et al., 2018). They train the network in a latent way by looking for keypoints that best relate the information from two views of the underlying object. For example, Suwajanakorn et al. (2018) enforce multi-view consistency, silhouette consistency, camera pose estimation, and sparsity constraints to supervise the network without using any images with 3D keypoint annotations. The multi-view consistency ensures that the keypoints inferred from two different views are consistent. The silhouette consistency ensures that the keypoints project inside the silhouette of the object. The camera pose estimation and sparsity constraints ensure that the inferred keypoints do not degenerate to a single location. However, Suwajanakorn et al. (2018) only focus on rigid objects, and exploiting similar constraints for articulated human body is an interesting future direction.

Multi-Person Multi-View 3D Pose Estimation

As discussed above, multi-view consistency constraints are very useful to train the network in an unsupervised way. Interestingly, such constraints can also be used in supervised scenarios, for example, as additional regularization. Exploiting such constraints in multi-person pose estimation from multi-view images is another interesting area to explore.

Multi-Person 3D Pose Estimation

Monocular multi-person 3D pose estimation is the most challenging problem in articulated human pose estimation. First, 3D pose estimation from RGB input is an ill-posed problem. Therefore, the existing methods (Rogez et al., 2017; Mehta et al., 2017b) resort to the estimation of root-relative 3D pose. However, the root-relative 3D pose is insufficient in this case, since we require a global understanding of the depths of the person. Hence, the absolute depth for each person has to be estimated, which, however, is a challenging task. Even for approximate solutions, we need to extract the structural information from the scene (*i.e.*, planes, object sizes, perspective and line angles) in addition to the 3D poses of the persons. The structural information can be obtained from indoor scenes, but becomes challenging in outdoor scenes with complex backgrounds. Third, the available dataset with accurate annotations of more than two persons per image are scarce (Elhayek et al., 2015; Mehta et al., 2017b), and only provide annotations for test-sets. Fourth, in the case of videos, we may also need to track the persons which increases the complexity even further. Only the recent work of Zanfir et al. (2018) addresses this problem in the correct and desired way, and present an approach to reconstruct the absolute 3D poses of multiple people. In Chapter 9, we also proposed a constrained normalization of the root-relative depths which allowed us to recover the scale normalized absolute pose of the hand. The same approach can also be extended to human pose. While such constraints

and the work of Zanfir et al. (2018) are the first steps toward this challenging problem, this is an interesting direction to pursue in the future.

Pose Estimation of Interacting Hands

In Chapter 9, we presented an approach for hand pose estimation where the hands are localized using a hand detector. While the approach achieved good results under occlusions and complex articulations, it fails when two hands become close to each other. For example, it does not work very well if both hands are interlaced. Even with RGB-D data, very few works have addressed this problem (Tzionas et al., 2016). The main challenges arise due to the substantial similarity between both hands, and addressing these challenges using only an RGB input is still an open question.

Multi-Person – Multi-Object Pose Estimation

So far in this thesis, we have focused only on the human pose estimation and human-human interactions. However, a significant fraction of our daily-life interactions consists of interactions with objects. Therefore, it is of great interest to also estimate the pose of the objects along with the human body poses. While there exist some early works toward this direction (Desai and Ramanan, 2012), they are only limited to one person and one object at a time. Tackling the problem in unconstrained environments with an unknown number of objects and persons is a mostly unaddressed topic.

Holistic Understanding of Humans

As discussed at the beginning of this thesis, to develop autonomous systems that can naturally interact with humans, we need a holistic understanding of the human body, hands, face, object interactions, and their activities. So far these problems have been addressed separately in the literature, and building a holistic methods which can simultaneously perform all these tasks in a single formulation is our ultimate goal. Joo et al. (2018), recently, tried to combine the complete human body in a single model, but they require multiple cameras, and the performance is still far from what is necessary for detailed reasoning in unconstrained scenarios. Interesting future work would include methods that can perform all these tasks using a unified approach in any unconstrained environment from an RGB input.

Bibliography

Georgios Tzimiropoulos Adrian Bulat. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on page 13.)

Ankur Agarwal and Bill Triggs. 3D human pose from silhouettes by relevance vector regression. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004. (Cited on page 21.)

Ankur Agarwal and Bill Triggs. Recovering 3D human pose from monocular images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006. (Cited on pages 21 and 110.)

Barbara E Ainsworth, William L Haskell, Stephen D Herrmann, Nathanael Meckes, David R Bassett Jr, Catrine Tudor-Locke, Jennifer L Greer, Jesse Vezina, Melicia C Whitt-Glover, and Arthur S Leon. 2011 compendium of physical activities: a second update of codes and met values. *Medicine & science in sports & exercise*, 43(8):1575–1581, 2011. (Cited on page 153.)

Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 120 and 122.)

M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 4.)

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. (Cited on page 18.)

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009. (Cited on page 9.)

Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Discriminative appearance models for pictorial structures. *International journal of computer vision*, 99(3):259–280, 2012. (Cited on pages 9 and 15.)

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages x, 34, 40, 47, 61, 62, 73, 74, 75, 76, 80, 103, 110, 111, 112, 114, 118, 126, 150 and 153.)

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. (Cited on page 19.)

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33 (5):898–916, 2011. (Cited on page 30.)

Vassilis Athitsos and Stan Sclaroff. Estimating 3D hand pose from a cluttered image. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. (Cited on page 20.)

Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. (Cited on page 110.)

L. Ballan, A. Taneja, J. Gall, L. Van-Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on page 130.)

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1-3): 89–113, 2004. (Cited on page 31.)

Dhruv Batra, Payman Yadollahpour, Abner Guzman-Rivera, and Gregory Shakhnarovich. Diverse m-best solutions in markov random fields. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on page 14.)

Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimation. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2017. (Cited on page 12.)

Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on page 109.)

Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in neural information processing systems*, pages 831–837, 2001. (Cited on page 9.)

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 1994. (Cited on page 42.)

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*, 2008(1):1–10, May 2008. ISSN 1687-5281. doi: 10.1155/2008/246309. URL http://jivp.eurasipjournals.com/content/2008/1/246309. (Cited on pages 36, 62 and 79.)

Liefeng Bo and Cristian Sminchisescu. Twin gaussian processes for structured prediction. *International Journal on Computer Vision*, 87(1):28–52, 2010. (Cited on pages 21, 110 and 126.)

Liefeng Bo, Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. Fast algorithms for large scale conditional 3D prediction. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. (Cited on pages 21 and 110.)

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 19, 114, 120 and 122.)

Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 12, 39 and 54.)

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 17, 23, 49, 51, 81, 82, 83, 84, 149 and 151.)

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a. (Cited on page 84.)

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017b. (Cited on page 18.)

Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on pages 11, 39 and 54.)

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011. (Cited on page 47.)

O. Chapelle and M. Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information Retrieval*, 2010. (Cited on page 135.)

James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *CVPR*, 2016. (Cited on pages 14, 54, 61, 75, 77 and 84.)

Ching-Hang Chen and Deva Ramanan. 3D human pose estimation = 2D pose estimation + matching. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 20, 118, 119, 121 and 124.)

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2017a. (Cited on pages 12, 16 and 83.)

W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *International Conference on 3D Vision*, 2016. (Cited on page 22.)

Xianjie Chen and Alan L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Conference on Neural Information Processing Systems (NIPS)*, 2014. (Cited on pages 11, 39, 49, 100, 101 and 149.)

Xianjie Chen and Alan L Yuille. Parsing occluded people by flexible compositions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 16, 40 and 54.)

Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 16.)

Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017b. (Cited on page 13.)

Anoop Cherian, Julien Mairal, Karteek Alahari, and Cordelia Schmid. Mixing Body-Part Sequences for Human Pose Estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages 14, 54, 61, 92, 100, 101 and 149.)

Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 91, 92, 102 and 103.)

Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on page 62.)

Sunil Chopra and Mendu R Rao. The partition problem. *Mathematical Programming*, 59(1-3): 87–115, 1993. (Cited on page 30.)

Vasileios Choutas, Philippe Weinzaepfel, Jerome Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 154.)

Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a. (Cited on page 112.)

Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017b. (Cited on page 12.)

CMU. Carnegie mellon university graphics lab: Motion capture database, 2014. URL `http://mocap.cs.cmu.edu`. `mocap.cs.cmu.edu`. (Cited on pages 110 and 116.)

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, and Guodong Zhang. Deformable convolutional networks. 2017. (Cited on page 152.)

M. Dantone, J. Gall, G. Fanelli, and Luc Van Gool. Real-time facial feature detection using conditional regression forests. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on page 96.)

Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Human pose estimation using body parts dependent joint regressors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. (Cited on pages 10, 12 and 75.)

Matthias Dantone, Christian Leistner, Juergen Gall, and Luc Van Gool. Body parts dependent joint regressors for human pose estimation in still images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 36(11):2131–2143, 2014. (Cited on pages xv, 8, 11, 91, 92, 93, 94, 95, 99, 100, 101 and 149.)

T. E. de Campos and D. W. Murray. Regression-based hand pose estimation from multiple cameras. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. (Cited on page 130.)

Martin de La Gorce, David J. Fleet, and Nikos Paragios. Model-based 3D hand pose estimation from monocular video. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2011. (Cited on page 19.)

Chaitanya Desai and Deva Ramanan. Detecting actions, poses, and objects with relational phraselets. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on pages 91 and 156.)

Andreas Doering, Umar Iqbal, and Juergen Gall. Jointflow: Temporal flow fields for multi-person pose tracking. In *British Machine Vision Conference (BMVC)*, 2018. (Cited on pages 151 and 152.)

Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 20 and 124.)

Marcin Eichner and Vittorio Ferrari. Better appearance models for pictorial structures. In *British Machine Vision Conference (BMVC)*, 2009. (Cited on page 10.)

Marcin Eichner and Vittorio Ferrari. We are family: Joint pose estimation of multiple persons. In *European Conference on Computer Vision (ECCV)*, 2010. (Cited on pages 15, 40, 61, 75 and 76.)

Marcin Eichner and Vittorio Ferrari. Appearance sharing for collective human pose estimation. In *Asian Conference on Computer Vision*, pages 138–151. Springer, 2012. (Cited on page 10.)

A. Elhayek, E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. (Cited on page 155.)

M. Everingham, S.M.A Eslami, L. Van Gool, C.K.I Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge: A retrospective. *International Journal on Computer Vision*, 2015. (Cited on page 47.)

Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 16, 49 and 51.)

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision*, 61(1):55–79, 2005. (Cited on pages ix, 2, 8, 9, 10, 11, 28, 92, 93 and 94.)

Pedro F Felzenszwalb and Daniel P Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012. (Cited on page 9.)

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2010. (Cited on page 92.)

Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. (Cited on pages 9 and 33.)

Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, 100(1):67–92, 1973. (Cited on pages ix, 8, 9 and 153.)

David A Forsyth, Okan Arikan, Leslie Ikemoto, James O'Brien, Deva Ramanan, et al. Computational studies of human motion: part 1, tracking and motion synthesis. *Foundations and Trends® in Computer Graphics and Vision*, 1(2–3):77–254, 2006. (Cited on page 7.)

Oren Freifeld, Alexander Weiss, Silvia Zuffi, and Michael J. Black. Contour people: A parameterized model of 2d articulated human shape. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. (Cited on pages ix, 8 and 15.)

Dariu M Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 73(1):82–98, 1999. (Cited on page 7.)

Mona Fathollahi Ghezelghieh, Rangachar Kasturi, and Sudeep Sarkar. Learning camera viewpoint using cnn to improve 3d body pose estimation. In *3D Vision (3DV), 2016 Fourth International Conference on*, 2016. (Cited on page 22.)

Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Deva Ramanan, Manohar Paluri, and Du Tran. Simple, efficient and effective keypoint tracking. In *ICCV PoseTrack Workshop*, 2017. (Cited on pages 81, 82, 84 and 86.)

Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 18, 19 and 151.)

G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 14, 54, 101 and 102.)

Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. Using k-poselets for detecting people and localizing their keypoints. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages 40 and 54.)

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. (Cited on page 42.)

Wenjuan Gong, Xuena Zhang, Jordi Gonzàlez, Andrews Sobral, Thierry Bouwmans, Changhe Tu, and El-hadi Zahzah. Human pose estimation from monocular images: a comprehensive survey. *Sensors*, 16, 2016. (Cited on page 7.)

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. 2016. (Cited on page 28.)

Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear body pose estimation from depth images. In *Joint Pattern Recognition Symposium*, 2005. (Cited on page 110.)

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 153.)

Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, 2009. (Cited on page 19.)

Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *IEEE Computer Vision and Patter Recognition*, 2018. (Cited on pages ix, 8 and 15.)

Simon Haykin et al. Cognitive radio: brain-empowered wireless communications. *IEEE journal on selected areas in communications*, 23(2):201–220, 2005. (Cited on page 9.)

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on pages 12, 16, 17, 19, 73, 81, 82, 84, 137 and 139.)

Kaiming He, Georgia Gkioxari, Piotr DollÃ¡r, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 16, 18, 23, 81, 82 and 84.)

T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 1996. (Cited on page 19.)

David Hogg. Model-based vision: a program to see a walking person. *Image and Vision computing*, 1(1):5–20, 1983. (Cited on page 7.)

Peiyun Hu and Deva Ramanan. Bottom-up and top-down reasoning with hierarchical rectified Gaussians. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on page 54.)

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara Balan, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 81 and 83.)

Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014. (Cited on page 95.)

Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 12, 17, 32, 39, 40, 49, 51, 54, 60, 67, 81, 83 and 148.)

Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Bjoern Andres, and Bernt Schiele. Arttrack: Articulated multi-person tracking in the wild. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 17, 18, 49, 51, 76, 77, 80, 81, 83, 149 and 151.)

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015. (Cited on page 13.)

C. Ionescu, Joao Carreira, and C. Sminchisescu. Iterated second-order label sensitive pooling for 3D human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014a. (Cited on pages 21 and 110.)

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014b. (Cited on pages xiii, 77, 110, 114, 124, 125 and 155.)

Umar Iqbal and Juergen Gall. Multi-person pose estimation with local joint-to-person associations. In *ECCV Workshop on Crowd Understanding*, 2016. (Cited on pages 4 and 54.)

Umar Iqbal, Martin Garbade, and Juergen Gall. Pose for action - action for pose. In *IEEE Conference on Automatic Face and Gesture Recognition*, 2017a. (Cited on pages 4 and 54.)

Umar Iqbal, Anton Milan, and Juergen Gall. PoseTrack: Joint multi-person pose estimation and tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017b. URL http://arxiv.org/abs/1611.07727. (Cited on page 4.)

Umar Iqbal, Andreas Doering, Hashim Yasin, Björn Krüger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D pose estimation in single images. *Computer Vision and Image Understanding (CVIU)*, 2018a. (Cited on page 5.)

Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via 2.5D latent heatmap regression. In *European Conference on Computer Vision (ECCV)*, 2018b. (Cited on page 5.)

Hamid Izadinia, Imran Saleemi, Wenhui Li, and Mubarak Shah. (MP)2T: Multiple people multiple parts tracker. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on page 18.)

Arjun Jain, Jonathan Tompson, Mykhaylo Andriluka, MPI Saarbruecken, Graham W Taylor, and Christoph Bregler. Learning human pose estimation features with convolutional networks. In *International Conference on Learning Representations*, 2014a. (Cited on page 11.)

Arjun Jain, Jonathan Tompson, Yann LeCun, and Christoph Bregler. Modeep: A deep learning framework using motion features for human pose estimation. In *Asian Conference on Computer Vision*, 2014b. (Cited on pages 14 and 54.)

Hueihan Jhuang, J. Gall, S. Zuffi, C. Schmid, and M.J. Black. Towards understanding action recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on pages 54, 61, 75, 77, 91, 92, 98 and 103.)

Sheng Jin, Xujie Ma, Zhipeng Han, Yue Wu, Wei Yang, Wentao Liu, Chen Qian, and Wanli Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, 2017. (Cited on pages 81, 82 and 84.)

Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973. (Cited on pages 3, 7 and 14.)

S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010a. (Cited on pages 74 and 75.)

Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010b. (Cited on pages 10, 48 and 61.)

Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2010c. (Cited on page 110.)

Sam Johnson and Mark Everingham. Learning Effective Human Pose Estimation from Inaccurate Annotation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. (Cited on pages 74 and 75.)

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 156.)

Shanon X Ju, Michael J Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, page 38. IEEE, 1996. (Cited on page 13.)

Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. (Cited on page 20.)

Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjorn Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on page 30.)

Ilya Kostrikov and Juergen Gall. Depth sweep regression forests for estimating 3D human pose from images. In *British Machine Vision Conference (BMVC)*, 2014. (Cited on pages 21, 110, 114, 118, 119, 123 and 126.)

Lynn T Kozlowski and James E Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Perception & psychophysics*, 21(6):575–580, 1977. (Cited on page 3.)

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. (Cited on page 11.)

Thorben Kroeger, Jörg H Kappes, Thorsten Beier, Ullrich Koethe, and Fred A Hamprecht. Asymmetric cuts: Joint image labeling and partitioning. In *German Conference on Pattern Recognition*, pages 199–211. Springer, 2014. (Cited on page 32.)

Björn Krüger, Jochen Tautges, Andreas Weber, and Arno Zinke. Fast local and global similarity searches in large motion capture databases. In *ACM SIGGRAPH Symposium on Computer Animation*, 2010. (Cited on page 113.)

Lubor Ladicky, Philip Torr, and Andrew Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. (Cited on pages 15 and 40.)

Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 19, 120 and 122.)

Yann LeCun, Léeon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *PROC. OF THE IEEE*, page 1, 1998. (Cited on pages 11, 25 and 28.)

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. (Cited on pages 26 and 28.)

Hsi-Jian Lee, CHEN Zen, et al. Determination of 3d human-body postures from a single view. *Computer Vision Graphics and Image Processing*, 30(2):148–168, 1985. (Cited on page 19.)

Evgeny Levinkov, Jonas Uhrig, Siyu Tang, Mohamed Omran, Eldar Insafutdinov, Alexander Kirillov, Carsten Rother, Thomas Brox, Bernt Schiele, and Bjoern Andres. Joint graph decomposition and node labeling: Problem, algorithms, applications. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 17, 32, 51 and 83.)

Sijin Li and Antoni B. Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, 2014. (Cited on pages 22, 110 and 124.)

Sijin Li, Weichen Zhang, and Antoni Chan. Maximum-margin structured learning with deep networks for 3D human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on pages 22 and 110.)

Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. (Cited on pages 36, 62 and 80.)

Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on page 12.)

Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3D pose sequence machines. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017a. (Cited on pages 22, 124 and 126.)

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014a. (Cited on pages 61, 110 and 111.)

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr DollÃ¡r, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014b. (Cited on pages 73, 74, 75, 76 and 78.)

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017b. (Cited on page 16.)

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 16 and 82.)

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. (Cited on pages ix, 26, 27, 28 and 29.)

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015. (Cited on pages 15, 19 and 23.)

Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. (Cited on page 19.)

Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. Lstm pose machines. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 14, 100, 101, 102 and 103.)

Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 22.)

A. Makris, N. Kyriazis, and A. A. Argyros. Hierarchical particle filtering for 3D hand tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on page 130.)

Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3D human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 21, 120, 122, 124 and 126.)

D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3D human pose estimation with a single RGB camera. In *SIGGRAPH*, 2017a. (Cited on page 23.)

Dushyant Mehta, Helge Rhodin, Dan Casas, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation using transfer learning and improved CNN supervision. In *http://arxiv.org/abs/1611.09813*, 2016. (Cited on page 124.)

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d body pose estimation from monocular rgb input. *arXiv preprint arXiv:1712.03453*, 2017b. (Cited on pages 23 and 155.)

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D human pose estimation with a single RGB camera. In *SIGGRAPH*, 2017c. (Cited on page 124.)

Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*, 2017. (Cited on pages 86 and 153.)

Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. (Cited on pages x, 37, 62, 63, 80 and 148.)

Thomas B Moeslund, Adrian Hilton, Volker Krüger, and Leonid Sigal. *Visual analysis of humans*. Springer, 2011. (Cited on page 7.)

F. Moreno-Noguer. 3D human pose estimation from a single image via distance matrix regression. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 21, 110, 118, 119, 120, 122, 124 and 126.)

Greg Mori and Jitendra Malik. Recovering 3D human body configurations using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006. (Cited on page 21.)

MPII-Leaderboard. MPII Pose Dataset Leaderboard. `http://human-pose.mpi-inf.mpg.de`, 2014. [Online; accessed 28-June-2018]. (Cited on page 7.)

F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 22, 136 and 138.)

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 2, 137, 141 and 143.)

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning (ICML)*, 2010. (Cited on page 26.)

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on page 136.)

Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 12, 13, 16, 17, 21, 28, 39, 54, 82, 112, 130, 134, 137 and 140.)

Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. (Cited on pages 17, 49 and 51.)

Bruce Xiaohan Nie, Caiming Xiong, and Song-Chun Zhu. Joint action recognition and pose estimation from video. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 92, 93, 98, 99, 100, 101, 102, 103 and 149.)

Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3D human pose estimation by predicting depth on joints. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 22 and 131.)

Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Generative partition networks for multi-person pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018a. (Cited on page 17.)

Xuecheng Nie, Jiashi Feng, Yiming Zuo, and Shuicheng Yan. Human pose estimation with parsing induced learner. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b. (Cited on page 13.)

M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on page 130.)

M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently creating 3D training data for fine hand pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on page 130.)

I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Markerless and efficient 26-DOF hand pose recovery. In *Asian Conference on Computer Vision*, 2010. (Cited on page 130.)

I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. (Cited on page 130.)

Joseph O'rourke and Norman I Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (6):522–536, 1980. (Cited on page 7.)

Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991. (Cited on page 32.)

P. Panteleris and A. Argyros. Back to RGB: 3D tracking of hands and hand-object interactions based on short-baseline stereo. In *arXiv preprint arXiv:1705.05301*, 2017. (Cited on page 130.)

Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *WACV*, 2018. (Cited on page 19.)

George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 16 and 83.)

George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *European Conference on Computer Vision (ECCV)*, 2018. (Cited on page 17.)

D. Park and D. Ramanan. N-best maximal decoders for part models. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. (Cited on pages 14, 15, 54, 92, 100, 101 and 102.)

Sungheon Park, Jihye Hwang, and Nojun Kwak. 3D human pose estimation using convolutional neural networks with 2D pose information. In *European Conference on Computer Vision Workshops*, 2016. (Cited on page 22.)

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 22, 110, 120, 122, 124, 126, 130, 131 and 134.)

Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018a. (Cited on pages 21, 110, 120, 122, 124, 126 and 154.)

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018b. (Cited on page 20.)

Christian Payer, Thomas Neff, Horst Bischof, Martin Urschler, and Darko Stern. Simultaneous multi-person detection and single-person pose estimation with a single heatmap regression network. In *ICCV PoseTrack Workshop*, 2017. (Cited on page 81.)

Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S. Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 2 and 13.)

DI Perrett, PAJ Smith, AJ Mistlin, AJ Chitty, AS Head, DD Potter, R Broennimann, AD Milner, and Ma A Jeeves. Visual analysis of body movements by neurones in the temporal cortex of the macaque monkey: a preliminary report. *Behavioural brain research*, 16(2-3):153–170, 1985. (Cited on page 3.)

T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015. (Cited on pages 14 and 54.)

Leonid Pishchulin, Arjun Jain, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Articulated people detection and pose estimation: Reshaping the future. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on pages 15 and 40.)

Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Proceedings of the IEEE international conference on Computer Vision*, pages 3487–3494, 2013a. (Cited on page 9.)

Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013b. (Cited on pages 11, 91 and 92.)

Leonid Pishchulin, Mykhaylo Andriluka, and Bernt Schiele. Fine-grained activity recognition with holistic and pose based features. In *GCPR*, 2014. (Cited on page 92.)

Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on pages xv, 17, 32, 35, 39, 40, 44, 45, 46, 47, 49, 53, 54, 60, 61, 62, 63, 80 and 148.)

Ralf Plankers and Pascal Fua. Articulated soft objects for video-based body modeling. In *IEEE International Conference on Computer Vision (ICCV)*, 2001. (Cited on page 19.)

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999. (Cited on page 47.)

Howard Poizner, Ursula Bellugi, and Venita Lutes-Driscoll. Perception of american sign language in dynamic point-light displays. *Journal of experimental psychology: Human perception and performance*, 7(2):430, 1981. (Cited on page 3.)

Gerard Pons-Moll, David J. Fleet, and Bodo Rosenhahn. Posebits for monocular human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages 20 and 154.)

Alin-Ionut Popa, Mihai Zanfir, and Cristian Sminchisescu. Deep multitask architecture for integrated 2D and 3D human sensing. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 22, 23, 110 and 126.)

PoseTrack-Leaderboard. PoseTrack Challenge Leaderboard. `https://posetrack.net/leaderboard.php`, 2018. [Online; accessed 28-June-2018]. (Cited on page 7.)

C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on page 130.)

Ibrahim Radwan, Abhinav Dhall, and Roland Goecke. Monocular image 3D human pose estimation under self-occlusion. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on page 126.)

U. Rafi, I.Kostrikov, J. Gall, and B. Leibe. An efficient convolutional network for human pose estimation. In *British Machine Vision Conference (BMVC)*, 2016. (Cited on pages 13, 39 and 54.)

Varun Ramakrishna, Takeo Kanade, and Yaser Ajmal Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *European Conference on Computer Vision (ECCV)*, 2012. (Cited on pages 20, 120 and 122.)

Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Tracking human pose by tracking symmetric parts. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. (Cited on pages 14 and 54.)

Varun Ramakrishna, Daniel Munoz, Martial Hebert, Andrew J. Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision (ECCV)*, 2014. (Cited on pages 10 and 12.)

Deva Ramanan. Learning to parse images of articulated bodies. In *Advances in neural information processing systems*, pages 1129–1136, 2007. (Cited on page 9.)

Deva Ramanan, David A Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 271–278. IEEE, 2005. (Cited on page 14.)

Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Adversarial collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. May 2018. URL `http://github.com/anuragranj/ac`. (Cited on pages 86 and 153.)

Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 16 and 138.)

James M. Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. In *European Conference on Computer Vision (ECCV)*, 1994. (Cited on pages 19 and 130.)

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Conference on Neural Information Processing Systems (NIPS)*, 2015. (Cited on pages 16, 19, 23, 41, 47, 64, 65, 67, 83 and 114.)

H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, B. Schiele H.-P. Seidel, and C. Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. In *IEEE Transaction on Graphics*, 2016. (Cited on page 110.)

Grégory Rogez and Cordelia Schmid. MoCap-guided data augmentation for 3D pose estimation in the wild. In *Conference on Neural Information Processing Systems (NIPS)*, 2016. (Cited on pages 22, 118 and 119.)

Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 22, 124 and 155.)

J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation*, 2010. (Cited on page 20.)

Javier Romero, Dimitirios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *Siggraph Asia*, 2017. (Cited on pages 19 and 130.)

Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 369–378. IEEE, 2017. (Cited on pages 19 and 34.)

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015. (Cited on page 137.)

R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3D hand pose reconstruction using specialized mappings. In *IEEE International Conference on Computer Vision (ICCV)*, 2001. (Cited on page 130.)

Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, 2004. (Cited on page 9.)

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986. (Cited on page 28.)

Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Bayesian image based 3D pose estimation. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on page 124.)

B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. (Cited on pages 74 and 75.)

Benjamin Sapp, Alexander Toshev, and Ben Taskar. Cascaded models for articulated pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 406–420, 2010. (Cited on page 11.)

Benjamin Sapp, David J. Weiss, and Ben Taskar. Parsing human motion with stretchable models. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. (Cited on pages ix, 8, 10, 13, 14, 34, 61 and 75.)

Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016. (Cited on page 7.)

Haoquan Shen, Shoou-I Yu, Yi Yang, Deyu Meng, and Alexander Hauptmann. Unsupervised video adaptation for parsing human motion. In *European Conference on Computer Vision (ECCV)*, 2014. (Cited on pages 14, 61 and 75.)

Yulong Shi, Xiaoguang Han, Nianjuan Jiang, Kun Zhou, Kui Jia, and Jiangbo Lu. Fbi-pose: Towards bridging the gap between 2d images and 3d human poses using forward-or-backward information. *ArXiv-Preprint*, 2018. (Cited on page 21.)

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011a. (Cited on page 110.)

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011b. (Cited on page 94.)

Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European conference on computer vision*, 2000. (Cited on page 19.)

Leonid Sigal, Alexandru Balan, and Michael J Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in neural information processing systems*, pages 1337–1344, 2008. (Cited on page 19.)

Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal on Computer Vision*, 87(1):4–27, 2010. (Cited on pages 77, 110, 114 and 155.)

Leonid Sigal, Michael Isard, Horst Haussecker, and Michael J Black. Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal on Computer Vision*, 98(1):15–48, 2012. (Cited on page 109.)

Edgar Simo-Serra, Arnau Ramisa, Guillem Alenyà, Carme Torras, and Francesc Moreno-Noguer. Single image 3D human pose estimation from noisy observations. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on pages 20, 114 and 126.)

Edgar Simo-Serra, Ariadna Quattoni, Carme Torras, and Francesc Moreno-Noguer. A joint model for 2D and 3D pose estimation from a single image. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. (Cited on pages 20, 123 and 126.)

T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 2, 34, 134, 136, 137, 140 and 143.)

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. (Cited on pages xii, 29, 73, 94, 95 and 149.)

Cristian Sminchisescu and Bill Triggs. Covariance scaled sampling for monocular 3d body tracking. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. (Cited on page 19.)

Cristian Sminchisescu, Atul Kanaujia, Zhiguo Li, and Dimitris N. Metaxas. Discriminative density propagation for 3D human motion estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. (Cited on pages 2 and 21.)

Jie Song, Limin Wang, Luc Van Gool, and Otmar Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 14, 100, 101 and 102.)

Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 143.)

S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *International Conference on 3D Vision (3DV)*, 2014. (Cited on page 130.)

S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on page 130.)

S. Sridhar, F. Mueller, M. Zollhöefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *European Conference on Computer Vision (ECCV)*, 2016. (Cited on pages 130, 136 and 141.)

Ke Sun, Cuiling Lan, Junliang Xing, Wenjun Zeng, Dong Liu, and Jingdong Wang. Human pose estimation using global and local normalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017a. (Cited on page 12.)

Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on pages 94, 96 and 97.)

X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on page 130.)

Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *IEEE International Conference on Computer Vision (ICCV)*, 2017b. (Cited on pages 12, 22, 110, 118, 119, 124, 130, 131, 137 and 139.)

Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *ArXiv-Preprint*, 2018. (Cited on page 155.)

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. (Cited on page 83.)

D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3D articulated hand posture. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on page 130.)

D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on page 130.)

S. Tang, B. Andres, M. Andriluka, and B. Schiele. Multi-person tracking by multicut and deep matching. In *ECCV Workshop on Benchmarking Multi-target Tracking*, 2016. (Cited on pages 30, 60, 61, 64 and 65.)

J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on page 130.)

Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3D human pose with deep neural networks. In *British Machine Vision Conference (BMVC)*, 2016a. (Cited on pages 22, 110 and 124.)

Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016b. (Cited on pages 22, 110 and 124.)

Bugra Tekin, Pablo Marquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2D and 3D image cues for monocular body pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 21, 22, 110, 120, 122 and 124.)

Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 21, 118, 119, 120, 122 and 124.)

J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. In *IEEE Transaction on Graphics*, 2014a. (Cited on page 130.)

Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Conference on Neural Information Processing Systems (NIPS)*, 2014b. (Cited on pages 11, 14, 39 and 92.)

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 39, 47 and 134.)

Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages 11 and 130.)

Hsiao-Yu Tung, Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Neural Information Processing Systems (NIPS)*, 2017. (Cited on page 20.)

D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal on Computer Vision (IJCV)*, 2016. (Cited on pages 130 and 156.)

Raquel Urtasun, David J. Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. (Cited on page 21.)

Srenivas Varadarajan, Parual Datta, and Omesh Tickoo. A greedy part assignment algorithm for real-time multi-person 2d pose estimation. *ArXiv-Preprint*, 2017. (Cited on pages 49 and 51.)

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 22 and 86.)

Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *European Conference on Computer Vision (ECCV)*, 2018. (Cited on page 20.)

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013. (Cited on page 77.)

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *European Conference on Computer Vision (ECCV)*, 2018. (Cited on pages 86 and 153.)

Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L. Yuille, and Wen Gao. Robust estimation of 3D human poses from a single image. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014a. (Cited on pages 20 and 126.)

Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on pages 102 and 103.)

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. (Cited on page 92.)

Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning, and recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014b. (Cited on pages 92 and 103.)

R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *IEEE Transaction on Graphics*, 2009. (Cited on page 130.)

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on pages ix, x, 12, 16, 17, 21, 28, 29, 30, 39, 42, 43, 47, 48, 54, 64, 65, 67, 100, 101, 102, 112, 114, 130, 134, 137, 140 and 143.)

Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on pages xi, 60, 61 and 151.)

Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017. (Cited on page 61.)

Ying Wu, John Y. Lin, and Thomas S. Huang. Capturing natural hand articulation. In *IEEE International Conference on Computer Vision (ICCV)*, 2001. (Cited on page 19.)

Fangting Xia, Peng Wang, Xianjie Chen, and Alan L. Yuille. Joint multi-person pose estimation and semantic part segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on page 18.)

B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, 2018. (Cited on pages 16, 19, 81, 82, 83, 84, 86 and 151.)

C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on page 130.)

Bin Yang, Junjie Yan, Zhen Lei, and Stan Z. Li. Convolutional channel features for pedestrian, face and edge detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on pages 94 and 149.)

Bo Yang and Ram Nevatia. An online learned CRF model for multi-target tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on page 62.)

Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on page 13.)

Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on page 22.)

Yi Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2013. (Cited on pages 15, 34, 91, 92, 99, 100, 101, 102 and 149.)

Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. (Cited on pages ix, 8, 10, 11, 14 and 20.)

Angela Yao, Juergen Gall, and Luc Van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal on Computer Vision*, 100(1):16–37, 2012a. (Cited on page 109.)

Angela Yao, Juergen Gall, and Luc Van Gool. Coupled action recognition and pose estimation from multiple views. *International Journal on Computer Vision*, 2012b. (Cited on page 92.)

Hashim Yasin, Björn Krüger, and Andreas Weber. Model based full body human motion reconstruction from video data. In *International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, 2013. (Cited on pages 20 and 112.)

Hashim Yasin, Umar Iqbal, Björn Krüger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D pose estimation from a single image. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. (Cited on page 5.)

Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2018. (Cited on pages 86 and 153.)

Fisher Yu, Vladlen Koltun, and Thomas A Funkhouser. Dilated residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on page 29.)

Tsz-Ho Yu, Tae-Kyun Kim, and Roberto Cipolla. Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. (Cited on page 92.)

Shanxin Yuan, Qi Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017. (Cited on page 150.)

Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes the importance of multiple scene constraints. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. (Cited on pages 23, 155 and 156.)

Dong Zhang and Mubarak. Human pose estimation in videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. (Cited on pages 14 and 54.)

J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3D hand pose tracking and estimation using stereo matching. In *arXiv preprint arXiv:1610.07214*, 2016. (Cited on page 136.)

Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on pages 54, 61, 75, 77, 93 and 98.)

Qiaoyong Zhong, Chao Li, Di Xie, Shiliang Pu, and Liang Ma. Towards realtime 2d pose tracking: A simple online pose tracker. In *ICCV PoseTrack Workshop*, 2017. (Cited on pages 81, 82 and 84.)

Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. 3d shape estimation from 2d landmarks: A convex relaxation approach. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. (Cited on pages 120 and 122.)

Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016a. (Cited on pages 22 and 124.)

Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Kostantinos G Derpanis, and Kostas Daniilidis. MonoCap: Monocular human motion capture using a cnn coupled with a geometric prior. *arXiv preprint arXiv:1701.02354*, 2017a. (Cited on pages 118, 119 and 124.)

Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *ECCV Workshops*, 2016b. (Cited on pages 2, 22, 110 and 124.)

Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Weakly-supervised transfer for 3D human pose estimation in the wild. In *IEEE International Conference on Computer Vision (ICCV)*, 2017b. (Cited on pages 22 and 131.)

Xiangyu Zhu, Yingying Jiang, and Zhenbo Luo. Multi-person pose estimation for posetrack with enhanced part affinity fields. In *ICCV PoseTrack Workshop*, 2017. (Cited on pages 81, 82 and 84.)

C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single RGB images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. (Cited on pages 2, 21, 34, 136, 137, 141 and 143.)

S. Zuffi, O. Freifeld, and M. J. Black. From pictorial structures to deformable structfures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. (Cited on pages ix, 8 and 15.)

Silvia Zuffi, Javier Romero, Cordelia Schmid, and Michael J. Black. Estimating human pose with flowing puppets. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. (Cited on page 54.)

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeiteinschlielich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie noch nicht veröffentlicht worden ist sowie, da ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen dieser Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Juergen Gall betreut worden.

Unterschift:

_____

Datum:

_____