MORE IS BETTER: 3D HUMAN POSE ESTIMATION FROM COMPLEMENTARY DATA SOURCES

DENIS TOMÈ

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY COLLEGE LONDON

THIS THESIS IS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

JANUARY 2021

DECLARATION:

I, Denis Tomè confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Denis Tomè

JANUARY 2021

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Prof. Lourdes de Agapito Vincente, for her valuable counsel and insight. I am very grateful for her support and involvement she has constantly demonstrated during our collaboration. I would also like to thank Dr. Chris Russell for his assistance and great support during the first year of Ph.D.

I would also like to sincerely thank the professors that inspired me during my entire education span, starting from G. Falcone during my high school years to Dr. Marco Tagliasacchi who supervised me during my MSc thesis. They deeply inspired me and made me enjoy what I do, a privilege I am very thankful for.

Furthermore, I would like to thank my parents, Mauro Tomè and Nadia Fumagalli, for giving me every ounce of support a person can ever ask. I am proud and honored to have them as parents; my friend and colleague Federico Monti for competing with me during our undergrad and post-grad studies in getting better marks, which made me push my boundaries beyond what I could have considered possible.

Finally, I would like to express my gratitude for my scholarship to the European Union, which every years allows many students like myself to follow a dream.

ABSTRACT

Computer Vision (CV) research has been playing a strategic role in many different complex scenarios that are becoming fundamental components in our everyday life. From Augmented/Virtual reality (AR/VR) to Human-Robot interactions, having a visual interpretation of the surrounding world is the first and most important step to develop new advanced systems.

As in other research areas, the boost in performance in Computer Vision algorithms has to be mainly attributed to the widespread usage of deep neural networks. Rather than selecting handcrafted features, such approaches identify which are the best features needed to solve a specific task, by learning them from a corpus of carefully annotated data. Such important property of these neural networks comes with a price: they need very large data collections to learn from. Collecting data is a time consuming and expensive operation that varies, being much harder for some tasks than others. In order to limit additional data collection, we therefore need to carefully design models that can extract as much information as possible from already available dataset, even those collected for neighboring domains.

In this work I focus on exploring different solutions for and important research problem in Computer Vision, 3D human pose estimation, that is the task of estimating the 3D skeletal representation of a person characterized in an image/s. This has been done for several configurations: monocular camera, multi-view systems and from egocentric perspectives.

First, from a single external front facing camera a semi-supervised approach is used to regress the set of 3D joint positions of the represented person. This is done by fully exploiting all of the available information at all the levels of the network, in a novel manner, as well as allowing the model to be trained with partially labelled data.

A multi-camera 3D human pose estimation system is introduced by designing a network trainable in a semi-supervised or even unsupervised manner in a multi-view system. Unlike standard motion-captures algorithm, demanding a long and time consuming configuration setup at the beginning of each capturing session, this novel approach requires little to none initial system configuration.

Finally, a novel architecture is developed to work in a very specific and significantly harder configuration: 3D human pose estimation when using cameras embedded in a head mounted display (HMD). Due to the limited data availability, the model needs to carefully extract information from the data to properly generalize on unseen images.
This is particularly useful in AR/VR use case scenarios, demonstrating the versatility of our network to various working conditions.

IMPACT STATEMENT

Human 3D pose detection for human-computer interaction is a very well known problem, part of the computer vision spectrum, which aims at identifying the 3D skeletal representation — which may vary based on the approach (E.g. positions, rotations, etc.) — of people from input images. Different variations of this task rely on different amount of available information to estimate the final pose. This ranges from complex systems made of multiple cameras with temporal consistency, to less complex ones relying for example on a single rgb camera set-up with frame-by-frame predictions.

As in other areas of computer vision, human 3d pose estimation is a task where deep learning approaches prevail in terms of accuracy in the estimations. However, these models require significantly larger datasets to learn from, to accurately and robustly work. The larger the data collection, the better. Yet, collecting the amount of necessary data is a non-trivial challenge which involves expensive equipment, properly calibrated, with an intense/time-consuming data preprocessing stage. This expensive and time consuming task is fundamental but few research groups have the resources to afford this.

Instead of limiting ourselves to the availability of data, we propose an alternative and less demanding solution which makes used of already existing and partially labeled data. Rather than relying on a single big dataset, we make use of a collection of complementary datasets, containing all the necessary information: dataset/s consisting of images with 2D annotations along with dataset/s consisting of 3D an-

notation only. Consequently, if needed, also the generation of new data becomes easier and less cumbersome.
These ideas and set of strategies can be translated to other research areas of computer vision for performance improvement, since introducing new data increase the robustness of the model towards variability of the input data.

As human 3D pose detection can be used in a variety of applications, we therefore explore also how this technology can be adapted to less common tasks of great use for augmented and virtual reality. We propose a novel and robust architecture which is able to couple with the larger amount of pose self-occlusion generated by using headset-mounted-cameras that is also able to achieve state-of-the art results on the normal front-facing-camera pose estimation task. This novel and unique work, is one of the first to open research towards a specific problem of AR/VR that if solved will push us closer to the futuristic idea of AR/VR we all have in mind.

CONTENTS

FIGURES **17**

TABLES **20**

NOMENCLATURE **22**

1 INTRODUCTION **24**

1.1 2D HUMAN POSE ESTIMATION 27

1.2 3D HUMAN POSE ESTIMATION 30

1.3 MACHINE LEARNING 35

1.3.1 LEVELS OF SUPERVISION 37

1.3.2 CLASSICAL ML 39

1.3.3 ARTIFICIAL NEURAL NETWORKS 40

1.4 THESIS STRUCTURE AND CONTRIBUTIONS 48

1.4.1 CHAPTERS 48

1.4.2 PUBLICATIONS 54

2 RELATED WORK **55**

2.1 MONOCULAR CONFIGURATION 58

2.1.1 2D POSE FROM IMAGES 58

2.1.2 3D POSE FROM KNOWN 2D JOINT POSITIONS 60

2.1.3 3D POSE FROM IMAGES 61

2.2 MULTI-VIEW CONFIGURATION 66

2.3 EXTERNAL CAMERA VIEWPOINT 68

2.3.1 FIRST PERSON CAMERA VIEWPOINT 68

2.3.2 Pose Estimation from Sensors 69

3 Pose from monocular image **70**

3.1 Overview 70

3.2 3D Pose detection framework 73

3.2.1 Probabilistic Model 74

3.2.2 2D to 3D pose inference 78

3.3 Datasets 86

3.3.1 Human3.6M 86

3.3.2 MPII and Leeds dataset 88

3.4 Experimental evaluation 89

3.4.1 Evaluation protocols 89

3.4.2 Quantitative Results 90

3.4.3 Qualitative Results 92

3.5 Conclusion 96

4 Pose from multi-camera views **97**

4.1 Overview 97

4.2 Multi-view framework 101

4.2.1 Architecture 103

4.2.2 3D pose estimation 105

4.3 Datasets 110

4.3.1 Human3.6M 110

4.3.2 CMU Panoptic Dataset 111

4.4 Data augmentation 114

4.4.1 Labeling data 115

4.4.2 Semi-supervised learning 118

4.5 Experimental evaluation 121

4.5.1 Evaluation protocols 121

4.5.2 Quantitative Results 122

4.5.3 Qualitative Results 128

4.6 Conclusion 131

5 Egocentric Human Pose Estimation **132**

5.1 Overview 132

5.1.1 Contributions 135

5.2 3D Pose detection framework 137

5.2.1 Architecture 138

5.2.2 2D Pose Detection 138

5.2.3 2D-to-3D Mapping 139

5.2.4 Pose rotation representation 142

5.2.5 Implementation details 143

5.3 Dataset 144

5.3.1 *x*R-EgoPose Dataset 144

5.3.2 *x*R-EgoPose^R Dataset 148

5.3.3 EgoCap dataset 148

5.3.4 Mo2Cap2 dataset 149

5.4 Experimental evaluation 154

5.4.1 Evaluation protocol 154

5.4.2 Quantitative Results on xR-EgoPose 155

5.4.3 Ablation studies 159

5.4.4 Results on egocentric real datasets 163

5.4.5 Evaluation on Front-Facing-Camera Datasets . . 166

5.4.6 Data-augmentation 167

5.4.7 Qualitative Results 168

5.5 Conclusion 174

6 Conclusions and Future Research **175**

6.1 Pose from monocular image 175

6.2 Pose from multi-camera views 177

6.3 Pose from egocentric perspective 178

6.4 Summary 179

APPENDICES **180**

A 3D LIFTER — GRADIENT PROPAGATION **181**

A.1 COMPUTING DERIVATIVES 181

A.1.1 LANDMARK GRADIENTS 182

A.1.2 MAPPING HM GRADIENTS 184

FIGURES

1.1 SKELETON COMPOSITION 28
1.2 2D LABELING 29
1.3 FISHER & ELSHLAGER: REFERENCE DESCRIPTION OF A FACE 29
1.4 FISHER & ELSHLAGER: IMAGE MATCHING EXPERIMENTS 30
1.5 SKELETON DEFINITION VARIABILITY 31
1.6 ILL POSED PROBLEM 32
1.7 HUMAN 3D LABELING 33
1.8 3D LABELING 35
1.9 MOTION CAPTURE MARKERS 36
1.10 BIOLOGICAL VS. ARTIFICIAL NEURON 40
1.11 FEATURE HYPERPLANE 42
1.12 FEED FORWARD NEURAL NETWORKS 43
1.13 ACTIVATION FUNCTIONS 44
1.14 CONVOLUTION OPERATION 45
1.15 RECEPTIVE FIELD 46
1.16 RECEPTIVE FIELD PIXELS 47
1.17 MONOCULAR POSE DETECTION 49
1.18 MULTI-VIEW 3D POSE DETECTION ARCHITECTURE 51
1.19 EGOCENTRIC 3D POSE ESTIMATION 52
1.20 EGOCENTRIC CAMERA PERSPECTIVE 52
1.21 EGOCENTRIC POSE ESTIMATOR ARCHITECTURE 53

2.1 Pupet representation 55
2.2 Tracking from multi-view 56
2.3 Tracking cyclic human motion 56

3.1 HYBRID ARCHITECTURE 73
3.2 POSE ALIGNMENT OF 3D POSES 77
3.3 ESTIMATIONS THROUGHOUT THE STAGES 79
3.4 EVOLUTION OF HEATMAPS THROUGH FUSION LAYER 81
3.5 JOINT PREDICTION REFINEMENT 84
3.6 HUMAN3.6M CAMERA POSITIONS 87
3.7 HUMAN3.6M ACTIONS 87
3.8 RESULTS ON IMAGES FROM THE MPII DATASET 93
3.9 RESULTS ON IMAGES FROM THE LEEDS DATASET 94
3.10 RESULTS FROM THE HUMAN3.6M DATASET 95

4.1 MULTI-VIEW CAMERA SET-UP 97
4.2 EXPLOITING GEOMETRY IN MULTI-VIEW 101
4.3 MULTI-VIEW ARCHITECTURE 103
4.4 HUMAN3.6M CAMERA POSITIONS 111
4.5 CMU PANOPTIC SYSTEM 112
4.6 CAMERA PLACEMENT IN CMU PANOPTIC DATASET 112
4.7 EXAMPLE IMAGES FROM CMU PANOPTIC DATASET 113
4.8 LABELING DATA WITH MULTI-CAMERA 3D POSE ESTIMATOR 115
4.9 MASK R-CNN OUTPUT 116
4.10 CROPPING REGION FROM SEGMENTATION MAP 118
4.11 MULTI-VIEW 3d POSE RECONSTRUCTIONS ON HUMAN3.6M DATASET . 129
4.12 SUPERVISED VS. SEMI-SUPERVISED MULTI-VIEW RECONSTRUCTIONS 129
4.13 SINGLE VS. MONOCULAR 130
4.14 LEVEL OF MISPREDICTIONS 131

5.1 EGOCENTRIC HUMAN POSE ESTIMATION PROBLEM 133
5.2 GENERATED EGO-HMD DATASET 134
5.3 CAMERA PERSPECTIVE FROM EGOCENTRIC POINT OF VIEW 135
5.4 EGOCENTRIC POSE ESTIMATION ARCHITECTURE 139
5.5 LATENT SPACE WITH SINGLE AND DUAL-BRANCH AE ARCHITECTURE 141

5.6 EXTENDED ARCHITECTURE 142
5.7 EXAMPLE FRONT FACING CAMERA DATASETS 144
5.8 XR-EGOPOSE CHARACTERS 145
5.9 XR-EGOPOSE DATASET CHARACTER HEIGHTS 147
5.10 EGOCAP CAMERA SETUP 148
5.11 EGOCAP VS. XR-EGOPOSE CAMERA VIEW 150
5.12 MO2CAP2 CAMERA SETUP 150
5.13 MO2CAP2 SAMPLED FRAMES 151
5.14 XR-EGOPOSE SHADOWS 153
5.15 MO2CAP2 VS. EGO-HMD 153
5.16 JOINT PDF OVER THE TEST-SET 158
5.17 HAND VISIBILITY 159
5.18 HM ESTIMATORS PERFORMANCE UNDER NOISE 161
5.19 HEATMAPS RECONSTRUCTION FROM THE DECODER BRANCH 169
5.20 CHARACTER ANIMATION 170
5.21 ANALYSIS OF THE ANGLE PREDICTIONS THROUGH TIME 170
5.22 QUALITATIVE RESULTS ON *X*R-EGOPOSE 171
5.23 QUALITATIVE RESULTS ON *X*R-EGOPOSE^R 171
5.24 QUALITATIVE EVALUATION ON MO2CAP2 DATASET 172
5.25 LATENT SPACE INSPECTION 173

TABLES

3.1 EVALUATION ON HUMAN3.6M DATASET USING PROTOCOL 1 91
3.2 EVALUATION ON HUMAN3.6M DATASET USING PROTOCOL 2 91
3.3 EVALUATION ON HUMAN3.6M DATASET USING PROTOCOL 3 92
3.4 EVALUATION OF PIXEL ERROR ON HUMAN3.6M DATASET 92

4.1 EVALUATION ON HUMAN3.6M USING PROTOCOL 1 123
4.2 EVALUATION ON HUMAN3.6M USING PROTOCOL 2 124
4.3 TWO CAMERA ONLY EVALUATION ON HUMAN3.6M DATASET 124
4.4 DATA AUGMENTATION EVALUATION ON MONOCULAR APPROACHES . 125
4.5 MODEL TRAINED ON UNLABELED DATA EVALUATED ON P1 126
4.6 MODEL TRAINED ON UNLABELED DATA EVALUATED on P2 127
4.7 MONOCULAR EVALUATION WITH DIFFERENT LEVELS OF LABELS . . 127
4.8 POSE ESTIMATOR VARIATIONS 128

5.1 XR-EGOPOSE PROPERTIES 147
5.2 XR-EGOPOSE EVALUATION 155
5.3 AVERAGE RECONSTRUCTION ERROR ON XR-EGOPOSE 157
5.4 HM REGRESSORS ANALYSIS 160
5.5 AVERAGE RECONSTRUCTION ERROR PER JOINT 162
5.6 AVERAGE RECONSTRUCTION ERROR BASED ON HM SIZES 162
5.7 MODEL EVALUATION BASED ON SKIN TONES 163
5.8 RECONSTRUCTION ERROR ON REAL DATA 164
5.9 COMPARISON AGAINST EGOCAP METHOD 164
5.10 EVALUATION ON MO2CAP2 DATASET 165
5.11 EVALUATION ON HUMAN3.6M DATASET 167

5.12 TRAINING SIZE: *X*R-EGOPOSE 168

5.13 TRAINING SIZE: FRONT FACING CAMERA 168

NOMENCLATURE

Roman Symbols

$\mathbf{b}$	Bias vector A Matrix
a	Vector
$\mathbf{W}$	Weight matrix

Greek Symbols

λ	Penalty weight
ϕ	Nonlinear function
θ	Parameters of a Neural Network model
Π	Projection
σ	Noise

Superscripts

(i)	Index
ij	Specific element of a matrix

† Pseudo-inverse matrix

Other Symbols

$\lvert\cdot\rvert$	Dimensionality of mathematical space
$\mathscr{L}$	Loss function
$\mathbb{R}$	Real number set
$\mathbb{Z}$	Integer number set

Acronyms / Abbreviations

AE	Autoencoder
ConvNet	Convolutional neural network
DeconvNet	Deconvolutional neural network
DAE	Denoising Autoencoder
e.g.	Exempli gratia (for example)
FC	Fully-Connected
i.e.	Id est (that is to say)
I	Input Image
Belief maps	Heatmaps
HM	Heatmaps
$\hat{HM}$	Predicted heatmaps
MSE	Mean Squared Error
NN	Neural Network
P	Human pose
$\hat{P}$	Pose prediction
PCA	Principal Component Analysis
ReLU	Rectified linear unit
Leaky ReLU	Leaky Rectified linear unit
SGD	Stochastic gradient descent
VAE	Variational Autoencoder
IRLS	Iterative Reweighted Least Squares
FOV	Field ov View
DoF	Degree of Freedom
NRSfM	Non-Rigid Structure from Motion
MoCap	Motion Capture

CHAPTER 1

INTRODUCTION

Computer Vision (CV) is concerned with the automatic extraction, analysis and understanding of relevant information from images and has been at the core of the new "automation revolution" providing tools to "perceive the world". It is a broad research field with areas ranging from robotic vision, where the interest is in studying image based techniques to allow robots to interact with and understand the world (such as safely navigating in an environment), to bio-medicine, where new algorithms are developed to assist doctors in making better diagnosis or assisting them in surgery tasks.

A well defined area of research in Computer Vision (CV) is *Human Pose Estimation*, which focuses on solving the problem of estimating the body configuration of one or multiple people from an image / multiple images / video sequences / etc. Specifically, from the input data (image, images, footage etc.) the goal is to find each of the poses characterizing the people represented in the data.

The pose definition varies according to the applications. For example it can be expressed as a set of 2D joint locations, 3D joint locations, joint angles as rotations relative to the parent node, etc.

Human Pose Estimation is a problem that has been researched in the community for more than four decades and despite the many years of research it is still considered

to be very challenging due to its large variability of conditions and characteristics. If we analyze the human anatomy, we can understand where part of the complexity comes from. A human body has a large number of muscles (over 600) that when flexed and moved change the appearance of the body, making it hard for a pure visual model the learn the large number of possible combinations. Moreover, we have 200 bones with over 200 joints, each increasing the capability of our body to bend and move in specific ways.

Additional variability is introduced by considering the various degrees of occlusion or self-occlusion that can occur when observing the body. Occlusion refers to the term for which only part of the body is directly visible from a camera perspective due to: *a)* objects being placed in front of limbs / areas of the body (occlusion), or *b)* due to the body (self-occlusion) where part of the body itself prevents us from observing other areas; for example, when crossing arms, only a portion of the arms is visible. Finally, humans also bulge, breath, flex and jiggle. Our shape changes with our age and fitness level and most importantly our visual appearance changes based on our outfit, the clothes materials and colors, which by itself introduces enormous variability.

Even if we only consider the factors described so far, without accounting for additional causes of complexity, one can easily understand the level of intricacy in solving this task and why it remains an open problem.

As in other areas of computer vision, machine learning (ML) algorithms have recently proven to outperform other types of solutions. However, when dealing with ML solutions, a discussion about data availability needs to be addressed as data collection is a challenging problem and machine learning models need to be carefully designed to deal with the scarcity of training data.

State-of-the-art approaches for human pose estimation introduced before the work presented in this thesis did not focus on the issue of how to exploit partially labelled datasets with models that are able to use the limited available information in a complementary way to improve the model performance. Specifically, *a)* end-

to-end approaches would only consider fully-labelled datasets where every input x would have an associated ground truth label y to be used for training, *b)* pipeline-approaches would use the partially labelled dataset in an independent manner. The model is trained to infer y in two independent steps going through intermediate futures z: $x \rightarrow z$ and $z \rightarrow y$. This would only partially tackle the problem of data availability because the model would not fully exploit interdependencies between the partial labels.

Without using datasets captured from different domains, the resulting solutions will *suffer major generalization issues*, with models not able to perform properly under different working conditions: e.g. different environments (indoor, outdoor, etc.), with dynamic lighting conditions, the different number of people, etc. Furthermore, this is particularly true when considering ML algorithms which are known to demand large datasets to be able to correctly model the solution.

Human Pose Estimation is an important research topic that has gained interest due to the complexity of the problem and the many challenges involved with solving it. There are many research topics in computer vision that require attention, however this specific task has proven to be particularly challenging to solve; with a large enough labelled dataset, every problem can be relatively solved with a pure learning strategy, however for Human Pose Estimation it is not possible to collect a corpus large enough to satisfy such requirements. Instead, the solutions need to be carefully designed to limit the problem and use the only available data to identify a solution.

There is a large abundance of applications that would benefit from such technology, from AR/VR to Robotics. It is a crucial step in the new ”advanced automation”, where systems need to be aware of people and know how to interact with them. For example, research is being developed to use human pose estimation to help elderly people being humanly unsupervised and autonomous in their homes, with a non-invasive AI system to monitor them and assist them when needed.

Many other applications lie in Augmented/Virtual reality and gaming applications.

For example, it plays a fundamental role in the new gaming concept of ”Metaverse”, a space where people virtually interact in a digital world that can be customized as pleased by the players to fit their creativity. In this scenario, human pose estimation is used for example to learn how people behave, look, move and interact with each other and to drive the avatars of the players.

THESIS OVERVIEW

This thesis is about designing machine learning models for the task of 3D human pose estimation that are able to use all the available information contained in the datasets. It is done to achieve better performance by exploiting the partially available labels during the model training stage, capitalizing on the data dependencies that would have not being otherwise exploited by previous state-of-the-art approaches. The set of solutions would facilitate the usage of different datasets, captured in different domains, to train models that as an effect would generalize better due to the variability of the input training data, as demonstrated in this thesis.

The remain of this chapter introduces the main concepts related to human pose estimation. Finally, I outline the thesis structure and I summarize the thesis contributions.

1.1 2D HUMAN POSE ESTIMATION

For 2D Human Pose Estimation we are going to solve this problem using machine learning. Here, the Convolutional Neural Network (CNN) model is predicting a pose as a collection of 2D joint positions, also called 2D skeletal representation.
Given a dataset of n pairs $\{(\mathbf{I}^{(i)}, \mathbf{P}^{(i)})\}_{i=1}^{n}$, we want to identify the *best* model θ that maps input images $\mathbf{I}^{(i)} \in \mathbb{R}^{S\times S\times 3}$ to their corresponding 2D poses $\mathbf{P}^{(i)} \in \mathbb{R}^{J\times 2}$ with $J \in \mathbb{Z}$ being the number of joints contained in the skeleton definition (see Fig. 1.1). Ground truth 2D joint positions $\mathbf{P}$ are usually annotated by users that manually

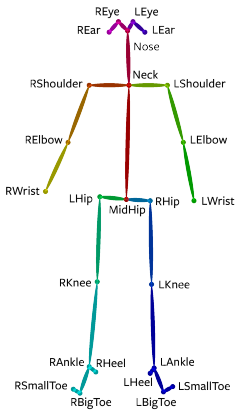


Figure 1.1: Example of skeleton definition from OpenPose [1].

label each joint in a sequence, frame by frame, using systems specifically designed to facilitate fast annotation. These annotations, although not perfect (as shown in Fig. 1.2), are fast to create and do not require any specific equipment.

Generic images downloaded from the internet are perfect candidates to be annotated. The ease of creating new data has enormous potential since it is possible, by carefully selecting which images to chose, to have a diverse dataset that if used during training allows a model to cope with a large variation of possible lighting conditions, clothes, etc. as well as different pose complexity with several levels of occlusions / self-occlusions.

The large number of available annotated datasets together with the large variability of human activities and conditions they cover, has allowed the community to develop accurate 2D joint estimation approaches. Current state-of-the-art approaches rely on deep convolutional neural networks, described later in this chapter, which have proven to be the best performing family of ML approaches.
However, the diversity of datasets has not been accompanied with a unified skeletal representation (see [2]). In practice, different definitions for datasets have introduced different definitions for skeleton which does not facilitate training of models across datasets (see Fig. 1.5 for example of different skeleton definitions).

Figure 1.2: Example of 2D joint annotations by humans (in red) compared against the resulting labels produced by [3] post-processing approach (in green). It is clear how annotations are not perfect, but due to the diversity a dataset and its size it is still valuable information.

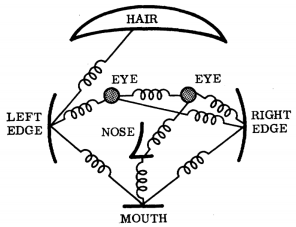


Figure 1.3: Fisher & Elshlager: Reference description of a face.

Not all approaches tackling 2D pose estimation however rely on ML solutions. For example in "*The representation and Matching of Pictorial Structures*" [4], Fisher and Elshlager defined a representation made of rigid pieces (components) held together by "springs", serving both as a constraint to the relative movement and a measure of cost of the movement by how much they are stretched (see Figure 1.3). Here, applying dynamic programming according to the algorithm described in the paper, they were able to run some image-matching experiments using faces as shown in Fig. 1.4.

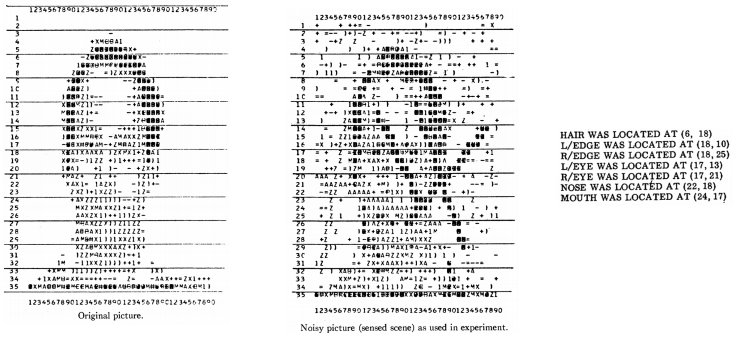


Figure 1.4: Fisher & Elshlager image matching experiment.

1.2 3D HUMAN POSE ESTIMATION

Similarly to the 2D pose estimation problem, we are going to focus exclusively on machine learning approaches for 3D human pose estimation. In order to describe the complications introduced by having 3D and for the sake of simplicity, we consider a scenario in which the model only uses a single input image to predict the 3D poses, represented as a set of 3D joint positions.

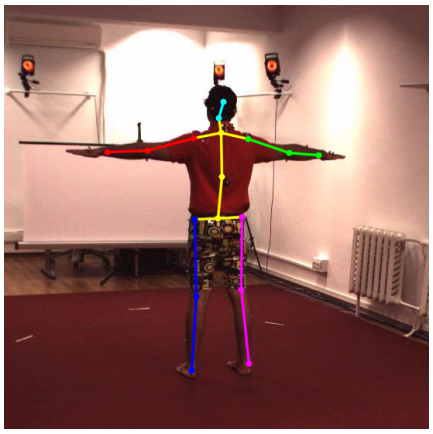
Given a dataset of n pairs $\{(\mathbf{I}^{(i)}, \mathbf{P}^{(i)})\}_{i=1}^{n}$, we want to identify the *best* model θ that maps input images $\mathbf{I}^{(i)} \in \mathbb{R}^{S\times S\times 3}$ to their corresponding 3D poses $\mathbf{P}^{(i)} \in \mathbb{R}^{J\times 3}$ with $J \in \mathbb{Z}$ being the number of joints contained in the skeleton definition (see Fig. 1.1).

The additional challenge arising with respect to the previous $Img \rightarrow 2D$ scenario is the concept of projections which in this case plays an important role: if we look at Figure 1.6a an illustration of the pinhole camera model is shown — with the image plane in front of the lens to simplify visualization — which can be used to describe the information loss coming from projections when acquiring images. In that diagram a 3D point $P = (X, Y, Z)$ is projected onto the image plane (screen), producing coordinate (x_s, y_s)

$$y_s = \frac{Y}{d} * f = -\frac{Y}{Z} * f \tag{1.1}$$

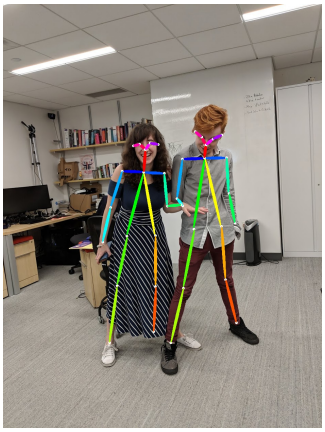
which can be described in matrix notation using homogeneous coordinates as

(a) COCO dataset

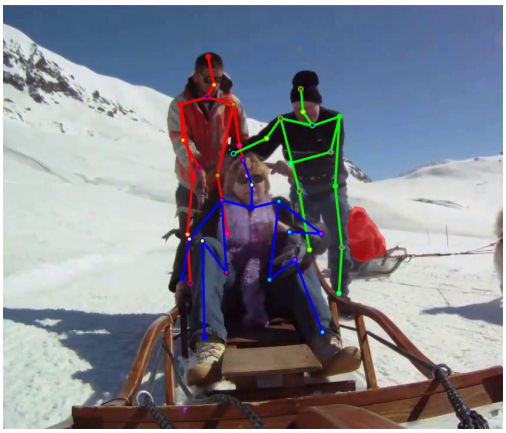


(b) Human3.6M dataset

(c) OpenCV model



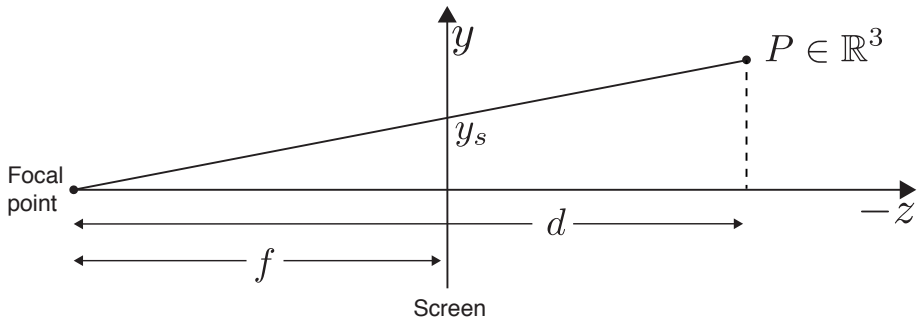
(d) OpenPose model



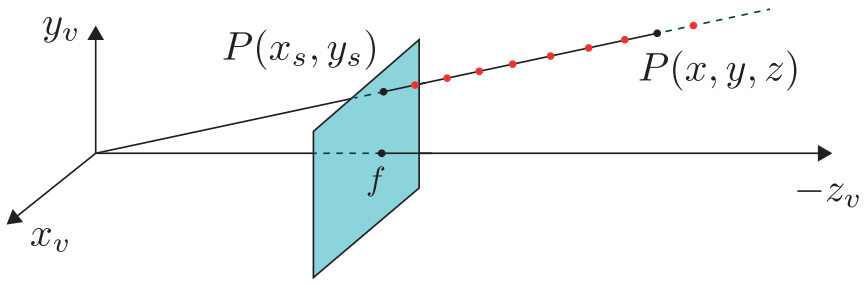
(e) PoseTrack dataset

Figure 1.5: Variability of skeleton definitions. This is an example of skeleton definitions according to different algorithms or datasets. The definition changes both in terms of number and joint positions.

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \Pi \cdot \mathbf{P} \tag{1.2}$$



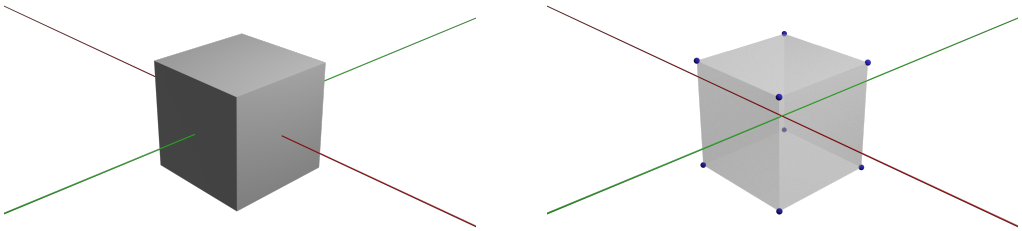
(a) Perspective Projection: estimating S_y point on the image plane from 3D position



(b) 3D from 2D keypoint: infinite number of possible solutions represented with red dots

Figure 1.6: Given a 2D point lying on the image plane, there is an infinite number of 3D positions that would satisfy the projection.

According to Eq. 1.2, the inverse operation of estimating a 3D point given its 2D corresponding coordinate results in an infinite number of possible solutions, as shown in Figure 1.6b. Therefore It is not possible to extract 3D information from input 2D data without defining additional constraints in the model.



(a) 3D Cube **(b)** 2D key-points on image

Figure 1.7: Human 3D labeling example.

As previously described, for machine learning based approaches, the data availability is a critical factor in determining a model that is able to learn the correct information to reliably perform under a different set of conditions, with good generalization. Unfortunately, unlike the 2D version of human pose estimation where image annotation is relatively easy, creating 3D pose labels results to be a very challenging task.

The main complication lies on the inability of humans to reliably and accurately estimate 3D locations. If we look at Fig. 1.7a, a person would be able to fairly reliably estimate the 2D key-points corresponding to the vertices of the cube as represented with blue spheres in Fig. 1.7b. However the person would perform extremely poorly in estimating their 3D corresponding positions, even in the case in which the cube dimensionality is known.

The process of 3D data annotation therefore requires a system capable of automatically annotate the data: a *motion capture*-like configuration. A motion capture studio is a specifically designed room containing a large number of cameras, from which it is possible to compute the 3D skeleton of a person. This technology relies on detecting small markers placed on the actor's body, which can then be used to geometrically compute their 3D position, knowing where the cameras were located. A visual representation of this is shown in Figure 1.8.

MoCap studios are currently used by industry, in cinematography as well as gaming, to animate virtual characters, mimicking the actor's performance. Although very accurate, these systems are very expensive, requiring constant and precise camera calibrations before each recording. Most notably, the biggest drawback of using such systems is in the constraints defined by this technology itself: whoever is captured is required to wear a motion capture suit (MoCap suit) as the actor represented in Fig. 1.8. As a result of this, the *variability* of the data captured in a MoCap studio is extremely limited due to the actor appearance as well as due to the *static background* which remains constant throughout the entire data capture sessions.
These constraints are such that a machine learning model purely trained on these data would not be able to generalize to "images in the wild" (images captured in the real-world — see Fig. 1.9b). Furthermore, even if dynamic backgrounds and different cloth combinations were used as a workaround for this problem, the markers (see Fig. 1.9) would still need to be visible in order for the system to work. These markers would introduce additional information absent on test data for which, again, the model would not be able to generalize well.

Due to these problems it is important to use diversified data captured under different conditions, from different domains, such that machine learning models train on those datasets are able to learn how to deal with a variety of different conditions and have better generalization performance.

In the next section, an exploration of basic machine learning techniques directly involved with the next chapters is introduced to better understand the concepts behind our model designs.

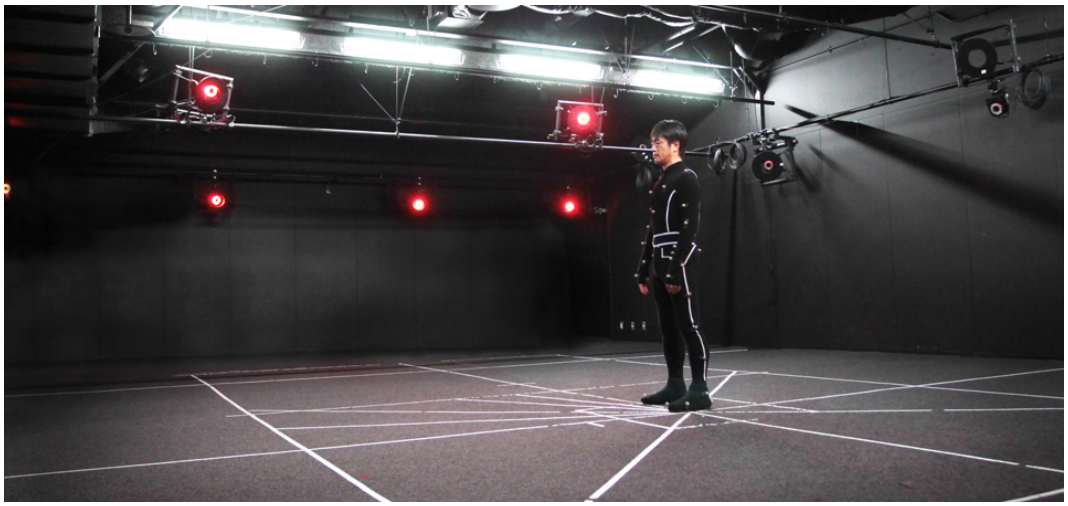


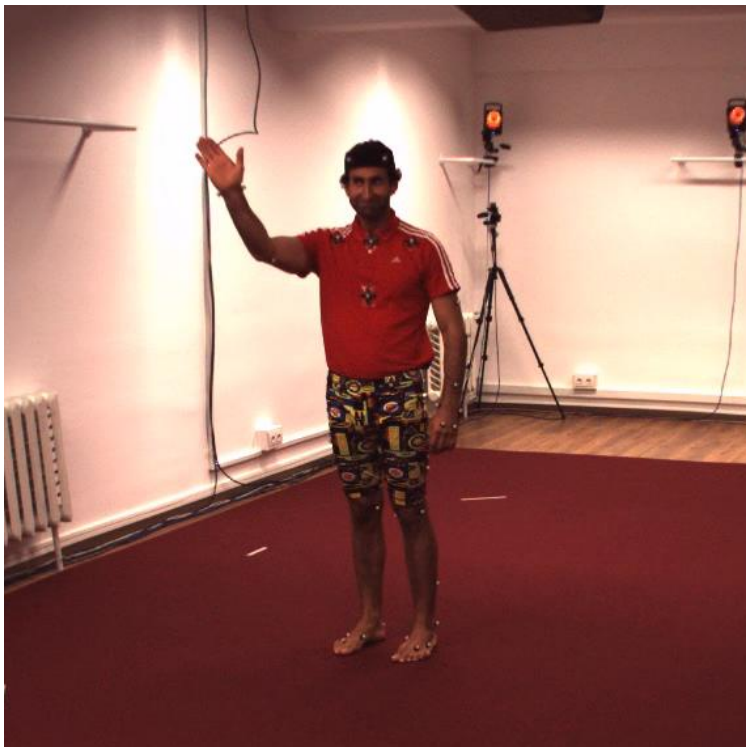
Figure 1.8: *Motion capture studio* showing various cameras capturing the actor while performing. The actor needs to wear a *mocap suit* (black dress) with markers (white dots) placed on it.

1.3 MACHINE LEARNING

The subject of ML is the study of mathematical models and algorithms that provide from the input data (training data) it receives the ability to make inferences and predictions without being explicitly programmed to do so. The widely accepted definition of what constitutes ML, given by Mitchell [5], is as follows:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E. In general, to have a well-defined learning problem, we must identify these three features: the class of tasks, the measure of performance to be improved, and the source of the experience."

With the *source of experience* being the observable input data for the defined *task*. Furthermore, input data may be provided with target output and based on the availability of output, ML model can be trained according to different levels of supervisions.



(a) Motion Capture image with actor wearing markers

(b) Realistic images we would need to train models

Figure 1.9: Top: image captured from a *motion capture studio* where markers can be seen on the actor. Furthermore, the entire dataset is captured in the same static environment. Bottom: real image we would ideally like to have with associated 3D skeleton information.

1.3.1 LEVELS OF SUPERVISION

Based on the available output information provided during the *training phase* of the model, different modalities of supervision emerge[1]: *supervised*, *semi-supervised*, *self-supervised* and *unsupervised*. The level of supervision is referred to the availability of data used for training the model.

The dataset is divided into sets with different functionality: *train-set)* the part of the dataset dedicated exclusively to train the model; *validation-set)* the part of the dataset used to evaluate the model during the training phase; and *test-set)* the part of the dataset dedicated exclusively to test the model; these data are unseen to the model during training or evaluation.

SUPERVISED

Supervised learning in ML is the task of learning a function/model that maps an input to an output, based on training $\{(I^{(i)}, y^{(i)})\}_{i=1}^{N}$ pairs.

Without constraining the description to any specific ML algorithm, if we consider a model θ, given input data I, ideally we want the model to predict the *known expected output* y. This could be described as

$$\arg\min_{\theta}(||y - f(I|\theta)||) \tag{1.3}$$

where we want to find a model for which the predicted output $f(I|\theta) = \hat{y}$ is as close as possible to the expected one y. An example of supervised learning approach is by Bogo *et al.* [6].

SEMI-SUPERVISED

Semi-supervised learning in ML is the task of learning a function/model that also makes use of unlabeled data for training. Typically, a small amount of labeled data

[1] Reinforcement and active learning have been omitted since they are outside the scope of this work

with a large amount of unlabeled data.

As in supervised learning, we are given a set of N independently identical distributed examples $\{(I^{(i)}, y^{(i)})\}_{i=1}^{N}$ with I input and y corresponding labels. Additionally, we are given U unlabeled examples $\{I^{(i)}\}_{i=N+1}^{N+U}$ with input data only.
Semi-supervised learning attempts to make use of this combined information to improve the performance of the model. An example of such approach is [7].

SELF-SUPERVISED

Self-supervised learning (inspired by Biology (see work by Gopnik *et al.* [8])) is a more recent form of training that unlike previous levels of supervision it requires less feedback to be given to the model and where the model is trained to predict any part of its input from any observed part.

The justification for self-supervision is that it is expensive to label data for new datasets and some areas are "supervision-starved" where annotations are hard to obtain. It is based on the idea that large availability of unlabeled data that could be exploited to generate better models.
Self-supervised learning is therefore a form of unsupervised learning where the data provides the supervision: retain part of the data and train the model to predict that. An example of such approach is [9].

UNSUPERVISED

Unsupervised learning is a type of self-organizing Hebbian learning in which it is possible to extract unknown patterns in datasets without preexisting labels: only input data is available and no corresponding target labels.

The goal for unsupervised learning is to model the underlying structure of distribution in the data in order to learn more about the data. K-Means [10] is an example of this type of approaches.

1.3.2 CLASSICAL ML

In classical ML, algorithms rely on hand-crafted feature-representations which are then used to solve the task. Such features are the result of complex feature engineering, in which an exploratory analysis is performed on the data in order to understand what are the important characteristics that need to be selected and passed on to the machine learning algorithm. Due to this selection process, these algorithms result to be easier to interpret and understand.

Hand-crafted features are the response to the well known problem of the *curse of dimensionality* [11], where the learning complexity increases exponentially with the dimensionality of data, resulting in the need to exponentially larger data collections for a larger number of dimensions. Consequently, by selecting features in a lower-dimensional space, fewer data samples are required to reach statistically stable results. This feature selection process however also reduces the predictive power of the system that compromises the algorithm's performance for the task.

Several techniques have been developed to manually or automatically select those features which contribute the most to the predicted variable or output. These set of techniques are part of the *feature selection* or *variable selection* process. A feature selection algorithm can be seen as the combination of a search technique for proposing new feature *subsets*, along with an evaluation measure which scores the different feature subsets. The most trivial and intuitive selection process would be an algorithm that tests each possible subset of features, finding the one which minimizes the error rate.

Ideally, we would prefer to design non-parametric learning algorithms which are capable of automatically learning the best set of features, tailored for solving a specific task: learning a feature extractor able to transform raw data into an appropriate representation.

1.3.3 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are computing systems inspired by *biological neural networks*, a population of neurons interconnected by synapses that carry out a specific function when activated.

The precursor of current Artificial Neural Networks is the *perceptron algorithm*, introduced by Frank Rosenblatt [12], in which the idea was to use mathematical models to mimic parts of neurons, such as dendrites, cell bodies and axons.
In the biological neuron, signals are received from dendrites and sent through the axon once enough signal is received; this signal can then be used by another neuron as input (see Figure 1.10). Some signals are more important than others and connections can become stronger or weaker. This can be translated into a function that receives as input a list of *weighted input signals* and outputs a signal if the sum reaches a certain threshold. This simple model is powerful enough to solve simple classification tasks, however a single layer of perceptrons alone is not able to solve non-linear classification problems.

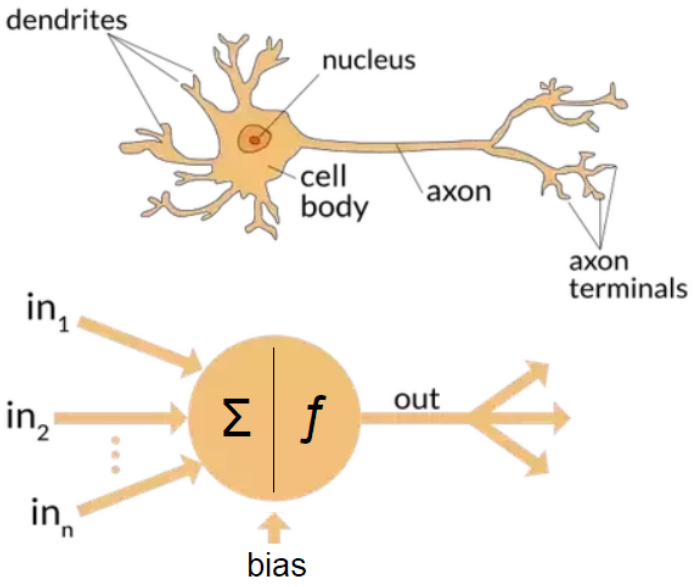


Figure 1.10: Mimicking biological neuron with an artificial one.
Image from `https://towardsdatascience.com`

This problem can be overcome by using multiple-layers of neurons, which has lead to the development of *Artificial Neural Networks* as we use them nowadays.

Mathematically, the perceptron can be described as a function that given some input $\vec{x} = x_1, x_2, \ldots, x_n$ produces an output variable o

$$o = f(w_1 * x_1 + w_2 * x_2 + \ldots + w_n * x_n) \tag{1.4}$$

with w_i the weights that have been learned by the network using an *activation function* f, defined as follows

$$f(x) = \begin{cases} 1 & \text{if } w_1 * x_1 + w_2 * x_2 + \ldots + w_n * x_n > b \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

where the linear combination of the input values or *features* is compared to a threshold value b called *bias*.

The problem with the Rosenblatt perceptron is in its expressive power: it only works with linearly separable data. To better visualize this, if we consider Equation 1.5 for a classification problem, all the elements with $\vec{w} \cdot \vec{x} > b$ (vector representation) would belong to class A, whereas if element has $\vec{w} \cdot \vec{x} \leqslant b$ it would belong to class B, as shown in the example in Figure 1.11b.

In this particular case it is possible to divide the space according to the hyperplane shown in Figure 1.11a. If the data we want to classify is not linearly separable, this approach is not usable to identify the best solution, and a more powerful model needs to be used: multi-layer neural networks.

Artificial Neural Networks, unlike classical machine learning approaches, learn the best set of features to be selected for a specific task to minimize the reconstruction error where, unlike the Rosenblatt perceptron, non-linearity can be used to solve the problem previously described.

FEEDFORWARD NEURAL NETWORKS

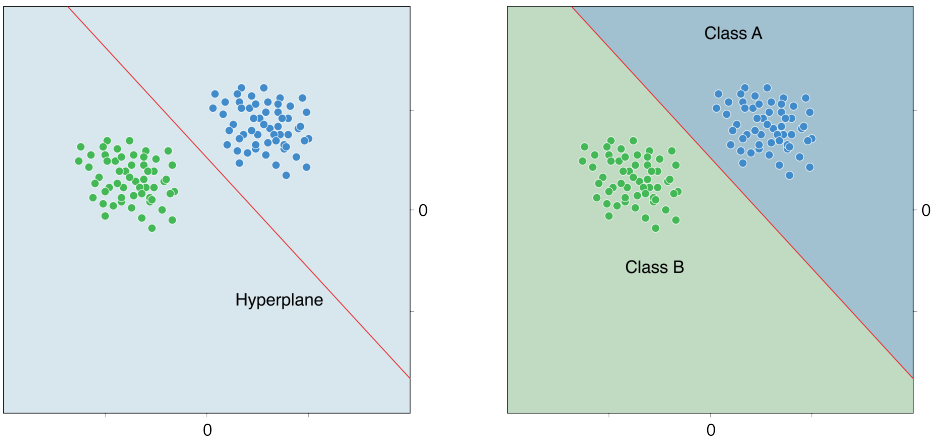
Feed-forward neural networks, also called multi-layer perceptrons, form the basis of many recently developed and more complex architectures. It is called "feed-forward" since the information always moves in one direction (from input layer to output layer) and never backwards, and each neuron in a layer has direct connection to the neurons of the subsequent layer, as shown in Figure 1.12.

In a feedforward neural network, Equation 1.4 is iteratively used to compute each neuron in the subsequent layer. Particularly, the output/activation a of a neuron is computed as

$$a_j^{(l)} = f\left(\sum_i w^{(i,j)} \cdot a_i^{(l-1)}\right) = f\left(\mathbf{W}^l \cdot a^{l-1}\right) \tag{1.6}$$

where $\mathbf{W}$ is the matrix containing the weights describing the connections between neurons. Note how in this formulation the bias $\mathbf{b}$ has been embedded in the matrix formulation (a value corresponding to each bias is 1). This operation is performed on all layers of the network, where the final output becomes

$$y = f\left(\mathbf{W}^L \cdot a^{L-1}\right) \tag{1.7}$$



(a) Hyperplane defined by $\vec{w}$ and b **(b)** Class assignment

Figure 1.11: Feature hyperplane

with output y defined as $y = [y_1, y_2]^T$.

During the *learning* stage of the network, the goal is to identify the model (set of weights connecting the neurons) such that the network is able to reproduce the desired output when the corresponding input information is provided. This can be mathematically described as minimizing a loss function

$$L(W) = \frac{1}{2}||y - \hat{y}||_2 \tag{1.8}$$

where y is the expected output and $\hat{y}$ is the output predicted by the model: network with the current weights $\mathbf{W}$. To produce more accurate results, the weights need to be properly adjusted.

The non-linearity of a neural network causes most interesting loss functions to become non-convex, which means that neural networks are usually trained by using iterative, gradient-based optimizers that merely drive the cost function to a very low value, rather than the linear equation solvers used to train linear regression models.

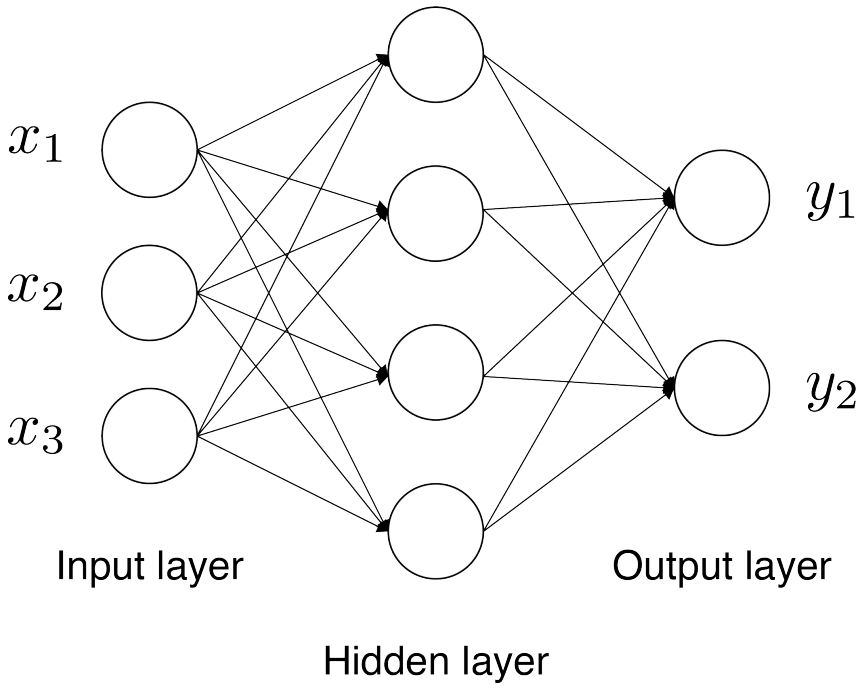


Figure 1.12: Feed forward neural network architecture

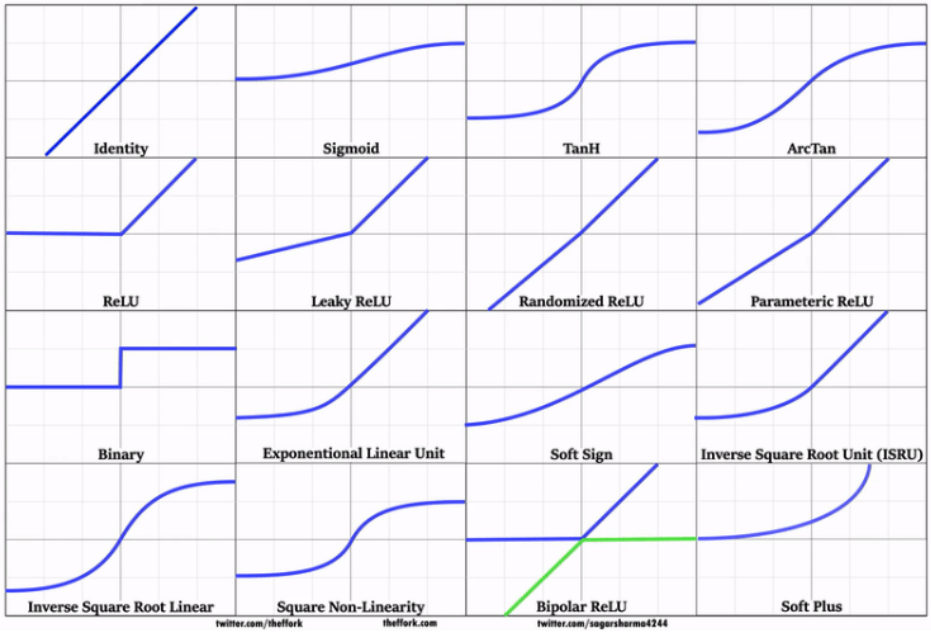


Figure 1.13: Activation functions

The learning process can be summarized by a sequence of steps [2]

1. **forward pass**: given input data x, the network predicts output $\hat{y}$ which produces error $\frac{1}{2}||y-\hat{y}||_2$

2. **backward pass**[3]: gradient computation to identify the "contribution" of each weight to the output error

$$\frac{\partial L}{\partial W^l}$$

3. **updating the model**: update the weights according to the gradient

$$w^{(l,t+1)} = w^{(l,t)} - \text{learning\_rate} * \frac{\partial L}{\partial W^{(l,t)}}$$

with t indicating the iteration step.

[2] In the simplest case of full-supervision: each input x is provided with corresponding label y

[3] more efficient mathematical formulations can be used to compute the gradients; such techniques will not be described as they are outside the scope of the thesis

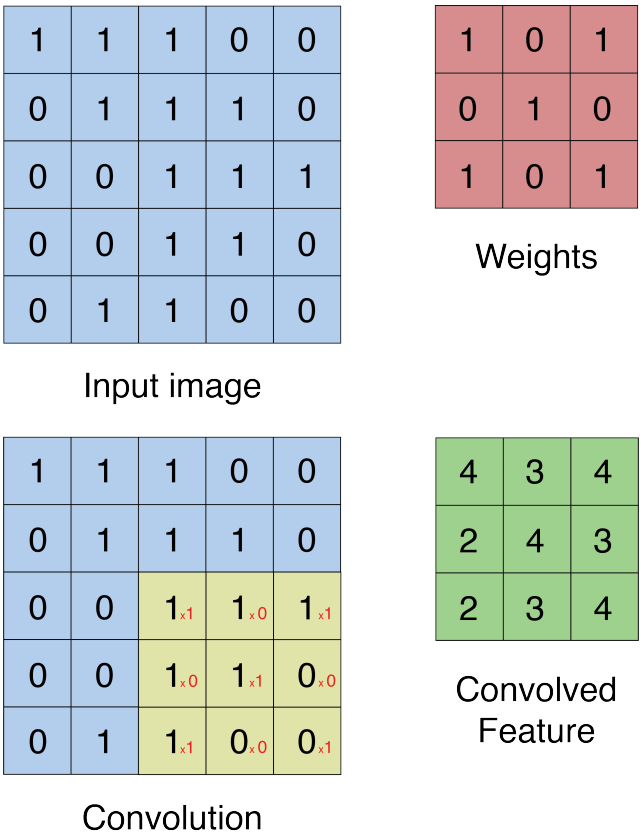


Figure 1.14: Convolution operation in 2D. Input image is convolved with the weights matrix resulting in the convolved feature matrix.

Activation function: the activation function described previously in Equation 1.5 when introducing the perceptron can be non-linear. Several functions have been used based on the task, as summarized in Figure 1.13.

CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks are a class of deep neural networks most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.

Convolutional Neural Networks is a specialization of feedforward nets where the hidden layers typically consist of a series of convolutional layers. This is the key-

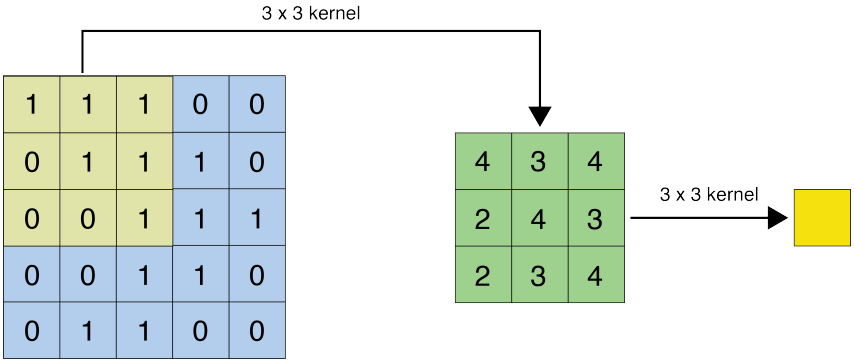


Figure 1.15: *Receptive field* across 3 different layers using 3×3 filters.

concept from which the name of the networks is derived from.

The convolution between two *1D continuous* functions f and g is defined as

$$(f * g) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\delta\tau \tag{1.9}$$

which, for a discrete case becomes

$$(f * g)[n] = \sum_{m=-M}^{M} f[n-m]g[m] \tag{1.10}$$

A visual representation of a discrete 2D convolution is shown in Figure 1.14, with the *Weights* matrix representing the set of weights that need to be learned by the network to maximise the output accuracy.

This new type of architecture exploits the *deepness* of the network to learn complex features able to generalize on different tasks. Furthermore, Convolutional Neural Networks introduce the concept of *receptive field*, which was firstly introduced by Alonso *et al.* [13] in biology as

> "The receptive field is a portion of sensory space that can elicit neuronal responses when stimulated"

The receptive field in CNNs refers to the region of the input space that affects a particular unit of the network. Note that this input region can be not only the input

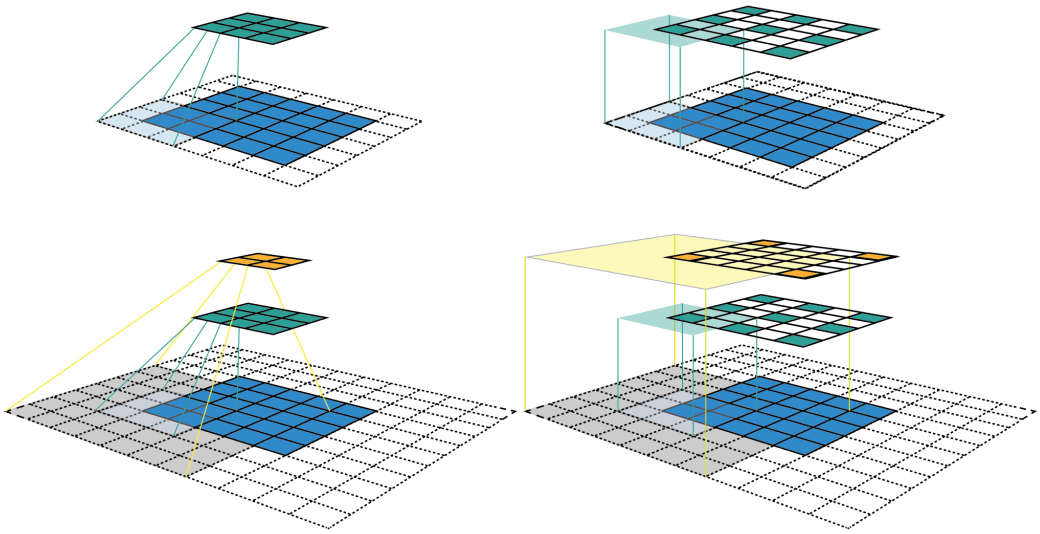


Figure 1.16: Convolution operation shown for two architectures. On the left, *feature map* visualization as previously represented; on the right, fixed size CNN *feature map* visualization and feature is located at the center of the receptive field. In both architectures the blue area represents the input image on which the convolution is performed on. If the first pixel of the image considered as the center of the convolution, it is possible to see the "visibility" of the operation over the information in the input as we move in the following *conv layers*, namely green and then orange. The deeper the CNN architecture, the more visibility we have over the input image when considering an output pixel of the convolution.

to the network but also outputs from other units in the network. Therefore the receptive field can be calculated relative to the input that is considered, and also relative the unit that is taken into consideration as the "receiver"of the input region (see Figure 1.15).

The receptive field of a feature can be described as its center location and size. Nonetheless, pixels in the receptive field are not equally important to the CNN's feature as, within a receptive field, the closer a pixel is to the center the more it contributes to the computation of the output feature. This means that a feature does not only look at a particular region of the input, but it also focuses exponentially more on the center of that region, as illustrated in [4] Figure 1.16.

[4]Source image taken from [14]

1.4 THESIS STRUCTURE AND CONTRIBUTIONS

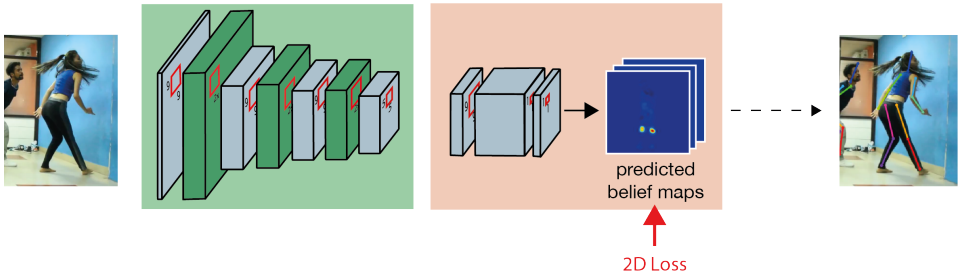
Based on Sections 1.1 and 1.2, it is clear that any pose estimation approach depends on the availability of training data due to the nature of the regressors that have been used: deep learning approaches. DL models have proven to perform better than standard non-DL algorithms (e.g. [4] vs. [15]), however the amount of data needed by them makes the problem even harder to solve due to the difficulties in data annotation and quality level previously described.

We argue that we must develop models that can manipulate data in a "smarter" and more efficient way, by not limiting ourselves to *fully labelled datasets* designed for the task (3D Human Pose Estimation), but rather to exploit relevant features that can be found in adjacent tasks we could benefit from to train better performing models and remove some working constraints currently being used (E.g. accurate 3D human pose labels in images in the wild).

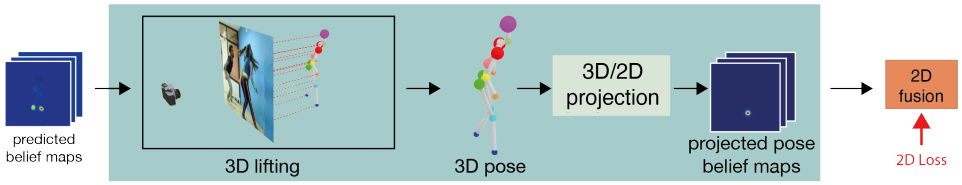
1.4.1 CHAPTERS

In Chapter 3 a novel semi-supervised convolutional neural network *hybrid-architecture* is presented, shown in Figure 1.17). With this CNN design we address the fact that some frames carry both 2D and 3D information, whereas some others only presents 2D labels. E.g. if a frame is not fully labelled, it is still valuable to use the provided information as a support for the model to improve the performance.

The architecture is classified as "hybrid" and it is composed by two *dependent* modules which share information but that are responsible to tackle sub-tasks of the overall problem. The module in Fig 1.17a) generates an initial estimation for 2D joint locations from the input image, expressed as a set of heatmaps — probability distribution of the joint location in the image — which can be quickly and easily translated into (u, v) pixel coordinates. The module in Fig 1.17b) takes the estimations generated by the previous module and relies on 3D information that has been mapped into a specifically constrained latent space to refine the estimation by



(a) Initial estimation



(b) Refinement using 3D

Figure 1.17: Deep neural network architecture for monocular 3D human pose estimation. It includes novel layers and it exploits 3D projection to train with limited 3D annotations by including 3D information in 2D predictions to refine the estimations.

injecting 3D constraints into 2D data, using a fusion layer.

The model is trained such that the data dynamically flows throughout the architecture, making the modules dependent.

This architecture has been defined as "hybrid" since it can be described as a solution in between two standard architecture designs:

a) **Pipeline architectures**: models made of a sequence of modules in which the output of one becomes the input to the consecutive one. This allows to have an *intermediate representation* of the data that can be controlled, visible as output of each module, that makes the model more flexible to train. Controlling the information propagated in the network however penalizes the performance, since the module's output futures are manually defined. E.g. To predict 3D poses from an input image, the first module predicts 2D joint locations from the RGB input data, followed by the second module which predicts 3D joint positions starting from the 2D joint information only; the *2D to 3D* module has no awareness of the RGB information that could potentially help generating better results.

b) **End-to-end architectures**: models are trained to regress the final output information from the input data without any intermediate handcrafted feature representation, but instead automatically learn the piece of information and its representation needed at each level of the architecture, to produce the best results. Approaches based on this method are less flexible to train on partially labeled datasets, since the network's internal representation is not legible.

The *hybrid architecture* introduced in this chapter is a hybrid between the two previously described designs: it has the flexibility of pipeline approaches — training the modules with the not fully labelled data — with the advantage of end-to-end approaches for which the modules are now dependent, exchanging information: the data flows throughout the entire network and the model has more flexibility in learning features that could be helpful in the consecutive modules. We prove how this novel method allows to use 3D information to improve 2D estimations, as well as achieving state-of-the-art results in 3D pose detection.

In Chapter 4 a more complex configuration is studied: a multi-view camera set-up where multiple images covering the recording area are jointly used generate more accurate predictions. The architecture presented in this work exploits some of the properties illustrated in the monocular counterpart, achieving similar flexibility. Specifically, the benefit of the newly-introduced "hybrid-architecture" — proven to perform well on monocular images — is extended to work in a different configuration. In this unique composition, the network can now exploit some geometrical constraints, due to the multi-view information, which consequently leads to better results. This novel architecture is presented in Fig. 1.18. Furthermore, this particular network design is flexible enough to allow us to further push the limit of data availability by permitting us to refine the model in an *unsupervised* manner.

It is shown how this novel multi-view architecture achieves *state-of-the-art* results on 3D human pose estimation on the most popular datasets. Moreover, we provide

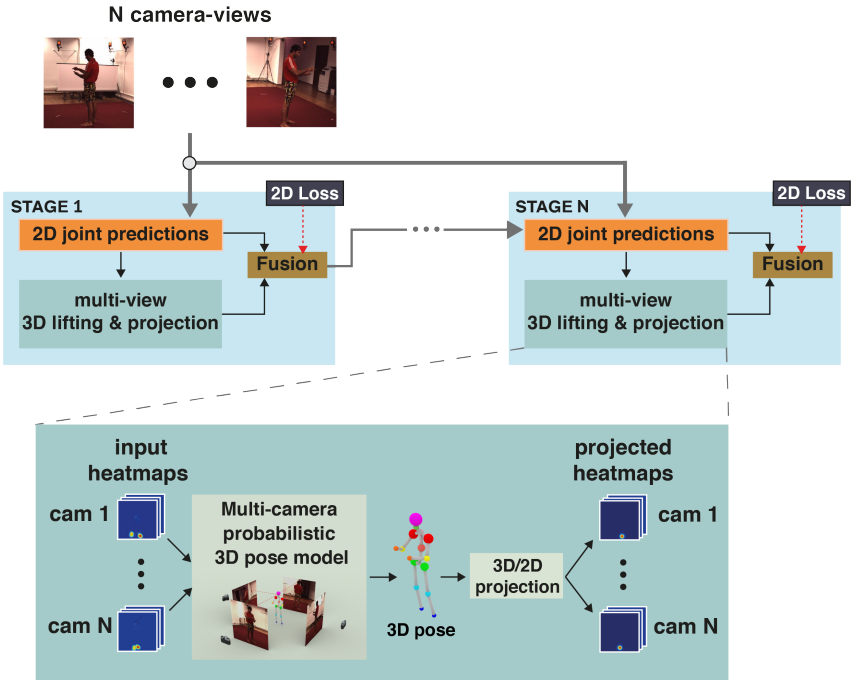


Figure 1.18: Deep neural network hybrid-architecture for 3D human pose estimation from multiple input images. It contains two modules: *a)* The initial estimator, producing 2D joint locations, and *b)* the refinement module which injects multi-view 3D pose constraints into 2D poses, by exploiting a *pose latent space* separately learned. Information is shared among the modules to guarantee better performance.

additional experimentation to show how such architecture can be used in a different configuration: as a*data annotator*. Given input images without associated labels, generate annotations using a learned model by producing qualitatively good labels that improve results of the methods trained with these new data.

Finally, Chapter 5 introduces a solution to a recent problem that has emerged within AR/VR research fields: 3D pose estimation from an *egocentric perspective* (see Figure 1.19).

Egocentric is used in this context to describe the camera perspective used for the images: the user is "wearing the camera" — either on an helmet or on the chest — which is then used to extract the required information. This is a relatively new problem that arose after the hardware miniaturization for which cameras could be mounted on VR/AR headsets or helmets in general. This new possibility offers

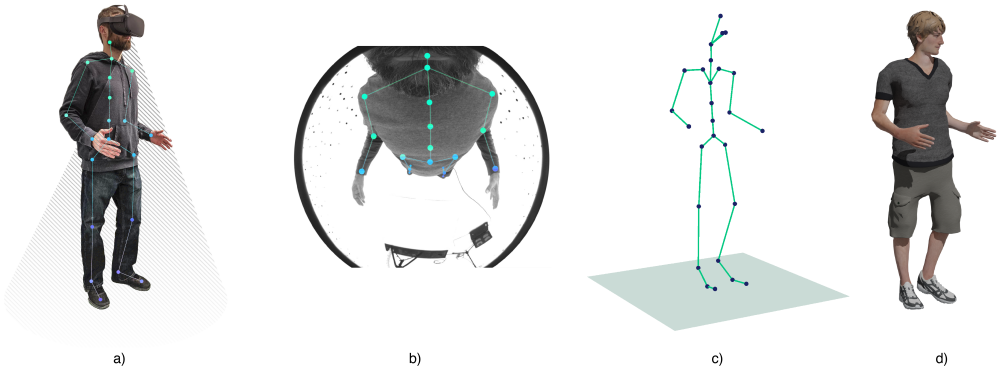


Figure 1.19: Egocentric 3D Human Pose Estimation where given as input image *b)*, representing pose *a)*, we want to identify the skeleton representation shown in *c)* which can be used for example to drive a virtual avatar as in *d)*.

freedom of movements to the user wearing the headsets without constraining them to a physical environment.

Due to nature of these unique camera perspectives (see Fig. 1.20), traditional front-facing camera datasets are not representative enough of real working camera perspectives to guarantee good reconstruction performance and generalization. This led to the *generation of a new, highly photo-realistic synthetic dataset*, publicly distributed to the community, that has proven to be effective for training models that can then transition to real world applications. It is orders of magnitude larger than the only other available dataset with dramatically higher photo-realism.

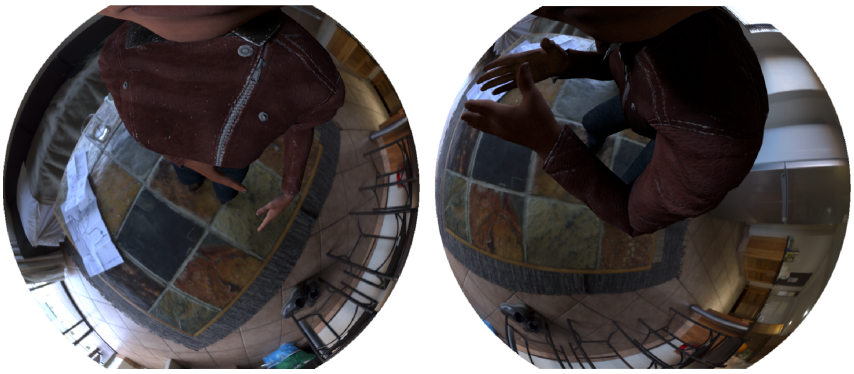


Figure 1.20: Egocentric Camera Perspective captured from a Headset Mounted Device (HMD) camera. Notice the level of occlusion of the lower body, characteristic of this task.

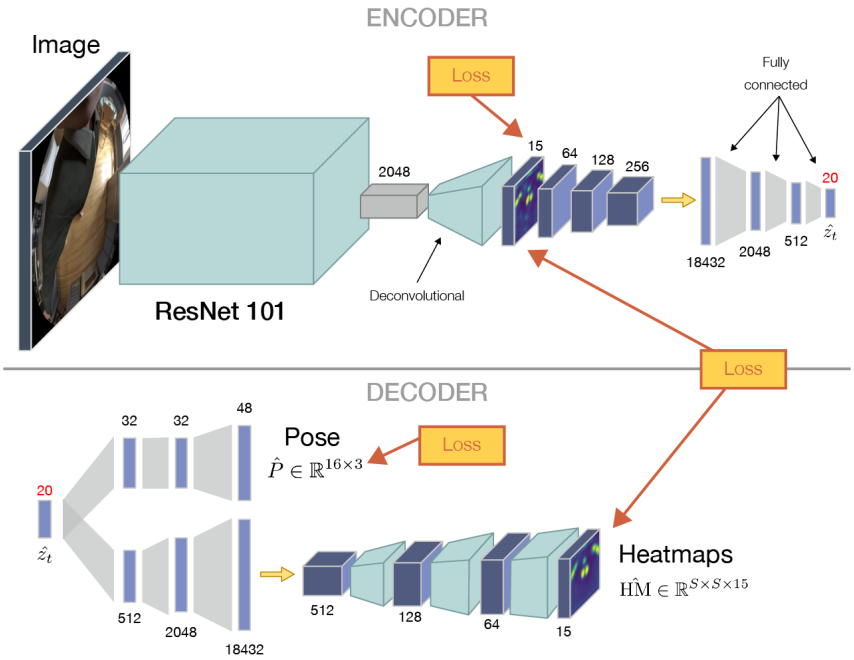


Figure 1.21: Architecture for 3D human pose estimation from an egocentric camera perspective. It relies on a multi-branch auto-encoder module which is able to compress the 3D poses with their associated prediction uncertainty in a latent space with good level of smoothness.

This dataset has been produced with tools and technologies used by VFX studios to produce the highest possible image quality and it has the equivalent size of two Pixar animated movies.

The peculiar camera perspective also proves to be challenging due to the large amount of self-occlusion — area of the body not visible due to been occluded by other areas — for which traditional CNN architectures for 3D human pose estimation would be ineffective. A novel architecture has been design (see Figure 1.21) to cope with such hard working conditions, generating poses with comparable reconstruction errors to those produced by typical front facing camera models.

This novel architecture makes use of an auto-encode architecture in a unique manner: using a multi-branch version. Having multiple branches allows us to make sure the latent space is including all the relevant information we want to be able to reconstruct, as well as the ability to rely on joint prediction uncertainties to learn how

the ambiguity of the joint locations can be properly minimized when accounting for 3D physically plausible poses.
We demonstrate how embedding joint uncertainty as well as locations is the key to this novel solution to work on such challenging conditions.

1.4.2 PUBLICATIONS

The work detailed in this thesis has contributed to the following publications

- **Denis Tome**, Chris Russell and Lourdes Agapito.
  Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. [16]
  IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

- **Denis Tome**, Matteo Toso, Lourdes Agapito and Chris Russell.
  Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture. [7] *International Conference on 3D Vision (3DV)*, 2018.

- **Denis Tome**, Patrick Peluse, Lourdes Agapito and Hernan Badino.
  xR-EgoPose: Egocentric 3D Human Pose from an HMD Camera. [17]
  Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.

- **Denis Tome**, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino and Fernando de la Torre.
  SelfPose: 3D Egocentric Pose Estimation from a Headset Mounted Camera. [18]
  IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2020

CHAPTER 2

RELATED WORK

3D Human pose estimation is quite a large area of computer vision that has been tackled for many years and it involves different specializations where the solution is found by using different levels of information.

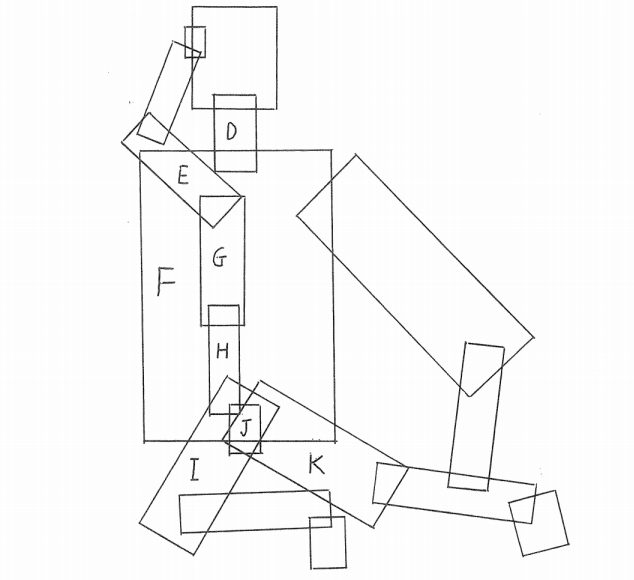


Figure 2.1: Pupet with representation defined in *G.E. Hinton* [19] with some extra rectangles. The idea solution only preserves the correct rectangles.

Since the first paper by *G.E. Hinton* [19] introduced almost 43 years ago (Fig. 2.1), many new approaches have been published, trying to find the right solution to deal with such a hard task. From generative methods, where the goal is to find a pose that matches the image data (edges, regions, color, textures, etc.), to approaches

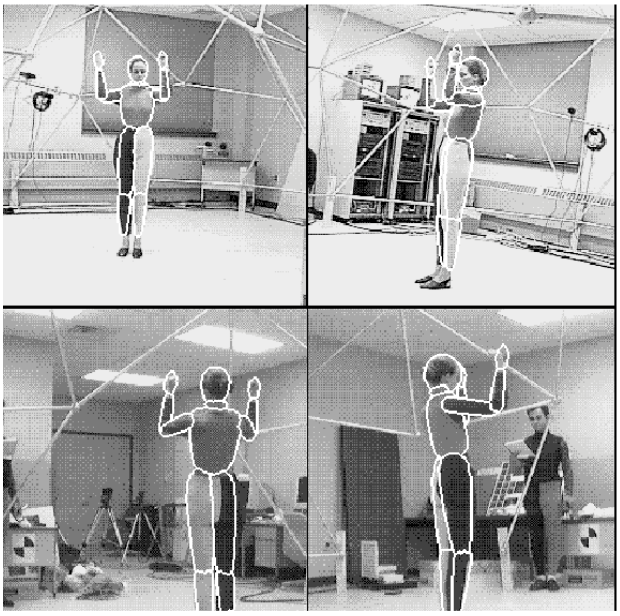


Figure 2.2: Tracking sequence results from Gavrila *et al.* [21] from four cameras (front, right, back and left) of a person.

that try to model poses using non-rigid structure from motion [20], or exploiting multi-camera setups systems, like the one proposed by Gavrila *et al.* [21] (Fig. 2.2). After not seeing any substantial improvement, new methods tried to improve the results by adding prior information, such as, Sidenbbladh & Black [22] (Fig. 2.3), Urtasun *et al.* [23] and even by using early deep neural networks [24].

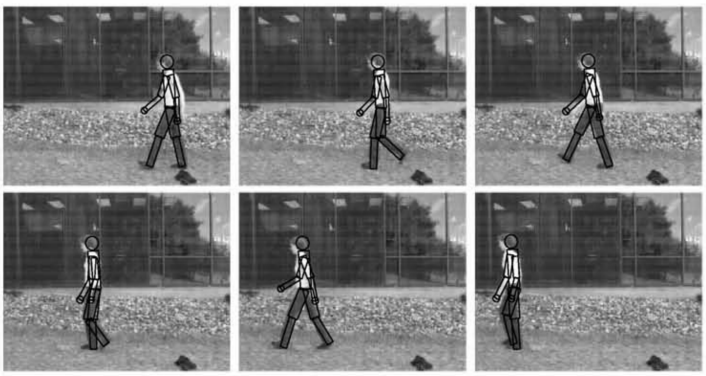


Figure 2.3: Tracking of a person walking, using Sidenbbladh & Black [22] showing results for frame 0, 10, 20, 30 , 40 and 50 with the projection of model configuration overlaid.

As in many other research fields, the introduction of deep learning led to a new era of research, where suddenly new promising results were achieved. From these nicer results, the interest of the community in this task has been constantly increasing which has led in the very last few years to new techniques producing astonishing results.

Some of these approaches rely on 2D pose estimations to later identify their projected 3D correspondences. The 2D visual recognition task of localizing body joints in the image is made difficult by multiple possible confusing factors including occlusion, variability in the color, shape and texture of clothing and the lighting conditions, while the task of lifting the 2D joint positions into 3D is even more challenging and intrinsically limited by the existence of perspective ambiguities. Due to this separation between 2D and 3D estimation, some important visual clues are lost and cannot contribute to the estimation of the 3D pose; however the intermediate representation, joint pixel position, is something that can be easily annotated by people, providing easy solutions for data augmentation to generate more robust feature detectors.

Other approaches directly use images to regress the 3D joint locations. In this architecture design, the model learns which are the most relevant features to exploit for better reconstruction performance. However, due to the nature of the information, no human-annotators can be used to label data and the set of images captured in specifically designed studios results in models performing poorly in real-world images.

In the next section it becomes evident how little emphasis has been given to the design of accurate models when the data availability is limited. Specifically, on how to deal with different data sources that when combined can be the solution to our data issues. Addressing this problem is however non trivial, mainly due to the different data representations used by the many available sources, as well as the challenges in designing a model that can be flexible enough in using at best the

information to achieve better results.

In this thesis we tackle the problem of how to deal with such small data availability to have models with good performance that is flexible enough towards training on multiple data sources.

This chapter is structured as follows: firstly I introduce the related work trying to solve 3D human pose estimation when using one or multiple cameras that are externally placed in the environment and face the person to be reconstructed (front-facing camera approaches). Secondly, I introduce a more recent set of methods describing a problem that has emerged with the advent of virtual and augmented reality where, due to agility and convenience reasons, cameras are placed on the person. This will introduce a new set of challenges for which the early approaches would not work.

2.1 MONOCULAR CONFIGURATION

Different approaches have been developed when a single RGB camera, pointing at the actor, is used to infer joint positions.

2.1.1 2D POSE FROM IMAGES

Pictorial structures: one of the classical approaches for articulated pose estimation, used by many methods [25, 26, 27, 28, 29, 30, 31, 32]. The idea is to infer body parts from local observations as well as to identify the correlation between the different parts. In order to express this, a tree-structured graphical model is used. Such methods proved to work well when all the limbs of the person are visible in the image. However, an effect that manifests itself with such approaches is also the double-counting of image evidence that occur due to correlations between variables that have not been captured by the tree-structured model. For example, if some parts have similar appearance, e.g. left and right arms or legs, the optimal score sometimes has them placed at the same position. This problem is especially prevalent in

2D where there is an inherent ambiguity, since two parts may project to the same image area even if they do not occupy the same volume in 3D.

Non-tree models: an extension of the previous approach used by [33, 34, 35, 36, 37], where the tree structure is augmented with additional edges to capture occlusions, symmetry and long-range relationships.

Hierarchical models: extension of pictorial structures model [38, 39] that represents the relationship between the different body parts at several sizes and scales, using a hierarchical tree structure. The idea behind this is that larger parts (e.g. limbs) usually have an image structure which is easier to detect then the single joints. Therefore, having this information would help in identifying smaller and harder to detect parts.

Deep convolutional architectures: with the introduction of DeepPose by Toshev *et al.* [40] research started to shift from the classical approaches to deep networks. In this case given an input image the $[x, y]$ coordinates of the joints are regressed using a standard convolutional architecture. A different approach, proposed by Tompson *et al.* [41] instead generates a heat-map for each of the joints by running the image through multiple resolution banks in parallel to simultaneously capture futures at different scales. It jointly uses ConvNet and a graphical model, which learns typical spatial relationships between joints. Other approaches are iterative or multi-stage training methods. Carreira *et al.* [42] uses the concept of Iterative Error Feedback, in which each run through the network takes as input both the image and the predictions from the previous run and further refines them. In similar spirit the work proposed by Wei *et al.* [15], Convolutional Pose Machines (CPM), builds on previous work of multi-stage pose machines [43], using ConvNets for feature extraction. Similarly to Carreira, each stage takes as input both the image and the heat-maps predicted at the previous stage, and performs a refinement operation in the predictions. Finally, Cao *et al.* [44] introduced a novel approach to efficiently

detect 2D poses of multiple people, simultaneously, in an image. This is possible due to the generation of additional information, part affinity fields, to learn how to associate body parts with individuals in the image.

2.1.2 3D POSE FROM KNOWN 2D JOINT POSITIONS

A large body of work has focused on recovering the 3D pose of people given perfect 2D joint positions as input. Early approaches [45, 46, 47, 48] took advantage of anatomical knowledge of the human skeleton or joint angle limits to recover pose from a single image. More recent methods [49, 50, 51] have focused on learning a prior statistical model of the human body directly from 3D mocap data.

Non-rigid structure from motion approaches (NRSfM) also recover 3D articulated motion [52, 53, 54, 55] given known 2D correspondences for the joints in every frame of a monocular video. Their huge advantage, as unsupervised methods, is they do not need 3D training data, instead of learning a linear basis for the 3D poses purely from 2D data. On other hand, their main drawback is their need for significant camera movement throughout the sequence to guarantee accurate 3D reconstruction. Recent work on NRSfM applied to human pose estimation has focused on escaping these limitations by the use of a linear model to represent shape variations of the human body. For instance, Cho *et al.* [56] defined a generative model based on the assumption that complex shape variations can be decomposed into a mixture of primitive shape variations and achieve competitive results on the CMU dataset. Since NRSfM methods use video data as input, temporal smoothness constraints can be used to improve performance.

Representing human 3D pose as a linear combination of a sparse set of 3D bases, pre-trained using 3D mocap data, has also proved a popular approach for articulated human motion [50, 57, 58]. While Zhou *et al.* [58] use a convex relaxation of the orthogonality constraint to convert the entire problem into a spectral-norm regularized least square problem, which is a convex program, [50] and [57] enforce

limb length constraints. Although these approaches can reconstruct 3D pose from a single image, their best results come from imposing temporal smoothness on the reconstructions of a video sequence.

There has been a lot of interest in the use of deep convolutional architectures for the task of articulated pose estimation, and recently Zhao *et al.* [59] has achieved state-of-the-art results by training a simple neural network to recover 3D pose from known 2D joint positions, where a new layer to solve the problem of occluded joints is introduced. While the results on perfect 2D input data are impressive, the inaccuracies in 2D joint estimation are not modeled and the performance of this approach combined with joint detectors is unknown.

2.1.3 3D POSE FROM IMAGES

Most approaches to 3D pose inference directly from images fall into one of two categories: *(i)* models that learn to regress the 3D pose directly from image features, and *(ii)* pipeline approaches where the 2D pose is first estimated, typically using discriminatively trained part models or joint predictors, and then lifted into 3D.
While regression based methods suffer from the need to annotate all images with ground truth 3D poses — a technically complex and elaborate process — for pipeline approaches the challenge is how to account for uncertainty in the measurements. Crucial to both types of approaches is the question of how to incorporate the 3D dependencies between the different body joints or to leverage other useful 3D geometric information in the inference process.

Many earlier works on human pose estimation from a single image relied on discriminatively trained models to learn a direct mapping from image features such as silhouettes, HOG or SIFT, to 3D human poses without passing through 2D landmark estimation [60, 61, 62, 63, 64]. A variety of learning frameworks such as random forests, regression, nearest neighbors, exemplars or SVMs have been used for this purpose.

Many recent direct approaches treat 3D pose estimation from a single input image as a fully supervised learning problem [65, 66, 41, 40]. Regression-based approaches make use of deep architectures to directly regress the 3D coordinates of human joints from the image [40, 65, 66, 67]. Much of the novelty of these approaches has involved combining end-to-end learning with expressive 3D priors to constrain the final 3D pose. Li and Chan [65] proposed strategies to jointly train for pose regression and body part detection. While [66] incorporate model joint dependencies in the CNN via a max-margin formalism, others [67] impose kinematic constraints by embedding a differentiable kinematic model into the deep learning architecture.

Tekin *et al.* [68] propose a deep regression architecture for structured prediction that combines traditional CNNs for supervised learning with an auto-encoder to learn a high-dimensional latent pose representation and which accounts for joint dependencies. Zhou *et al.* [67] enforce bone lengths in predictions. Tekin *et al.* also leverage 2D image data [69] by adding a second network stream whose outputs are fused with the 3D regressor.

Following the trend in 2D human pose estimation to predict heatmaps rather than regressing 2D landmarks, Pavlakos [70] predicted per-voxel likelihoods, or 3D heatmaps, for each joint using a coarse-to-fine approach.

Rogez *et al.* [71] proposed an end-to-end architecture that combines a region proposal network for human localization with classification and regression branches for joint estimation of 2D and 3D human pose. Sun *et al.* [72] adopted a bone based representation for the pose and propose a unified setting for 2D and 3D pose estimation that encoded long range interactions between bones. Both approaches achieve best results when a 2D loss is combined with the standard 3D loss. Zhou *et al.* [73] shared common representations between the 2D and the 3D tasks inside the network which is trained end-to-end with both 2D and 3D losses.

These methods share the disadvantage of generalizing poorly to images in the wild:

the need for ground truth 3D poses to train the image to 3D pose regressor means that they must be trained exclusively on images captured in MoCap studios, with all the limitations that come with it.

As CNNs have become more prevalent, 2D joint estimation [15, 74, 75] has become increasingly reliable and many recent works have looked to exploit this using a pipeline approach. Papers such as [76, 77, 41, 78] first estimate 2D landmarks and later 3D spatial relationships are imposed between them. The task becomes lifting the 2D coordinates into 3D either by model fitting [50, 51, 79, 80, 6, 81] or regression [82, 83, 84].

Simo-Serra *et al.* [85] was one of the first to propose an approach that naturally copes with the noisy detections inherent to off-the-shelf body part detectors by modeling their uncertainty and propagating it through 3D shape space while guaranteeing that geometric and kinematic 3D constraints were satisfied. The work proposed by Sanzari *et al.* [81] estimates the location of 2D joints using a state-of-the-art approach before predicting 3D pose using appearance and the probable 3D pose of discovered parts using a hierarchical Bayesian non-parametric model.

Zhou *et al.* [86] tackles the problem of 3D pose estimation for a monocular image sequence integrating 2D, 3D and temporal information to account for uncertainties in the model and the measurements. The uncertainty in the 2D joint estimates, predicted by a regression-based CNN, is marginalized out by an EM algorithm that lifts 2D poses into 3D while imposing 3D spatial relationships between joints via a pre-learnt sparsity-driven 3D geometric prior and temporal smoothness. Similar to our proposed approach, Zhou *et al.*'s method [86] does not need synchronized 2D-3D training data, *i.e.* it only needs 2D pose annotations to train the CNN joint regressor and a separate 3D mocap dataset to learn the 3D sparse basis; it however relies on temporal smoothness for its best performance, and poorly estimated human pose from a single image.

Moreno-Noguer [83] estimated 3D pose from 2D inputs using 2D-to-3D distance matrix regression. Chen and Ramanan [87] estimated the depth of 2D landmarks by matching them to a library of 3D poses. Bogo *et al.* [6] fitted a dense statistical shape and pose model, trained on thousands of 3D scans [88], to 2D joints obtained with DeepCut [75]. Martinez *et al.* [82] shows how even a simple regressor — a feed-forward network with residual connections and batch normalization — vastly outperforms previous approaches when given ground truth 2D landmarks as input, suggesting that the largest source of errors in 3D pose reconstruction is incorrect 2D estimation.

Sarandi *et al.* [89] generates volumetric heatmaps per body joint, which are converted to coordinates using soft-argmax and are used together with absolute person-center depth to produce the final 3d coordinates. Luvizon *et al.* [90] proposes a multitask framework for jointly 2D and 3D pose estimation from still images and human action recognition from video sequences, by showing that a single architecture can be used to solve the two problems in an efficient way and still achieves state-of-the-art results. Kanazawa *et al.* [91] achieve good reconstruction results by producing a richer and more useful mesh representation that is parameterized by shape and 3D joint angles.

Recently, Pavlakos *et al.* [92] proposed a solution to alleviate the need for accurate 3D ground truth by proposing to use a weaker supervision signal provided by the ordinal depths of human joints. This information can be acquired by human annotators for a wide range of images and poses. Furthermore these annotations can be easily incorporated in the training procedure of typical ConvNets for 3D human pose with results improvements on the model performance.

Training with 2D-only loss: A few approaches bypass the need to annotate images with 3D ground truth labels by keeping an internal 3D representation of the pose but training based on 2D re-projection losses. These approaches benefit from both

their ability to generalize to in-the-wild images as they do not rely on 3D annotated images that can only be captured in studios; and the added structural 3D pose priors afforded by internal 3D representation. Tome *et al.* [16] proposed a multi-stage architecture that reasons jointly about 2D and 3D pose to improve both tasks. Key to their architecture is a 3D lifting module that reconstructs 2D estimated landmarks in 3D and projects them back into 2D, as their end-to-end training minimizes deviations of the re-projected 3D landmarks from the ground truth 2D labels. Wu *et al.*'s single image 3D interpreter network [93] also uses a loss based on the 2D re-projection error of predicted 3D landmarks, along with a supervised 2D landmarks to 3D pose regressor. Tung *et al.* [94] combine a similar 2D re-projection loss with an adversarial loss and later [95] propose to combine strong supervision from synthetic data with a self-supervised loss based on consistency checks against 2D estimates of keypoints, segmentation and optical flow.

Yang *et al.* [96] proposes an adversarial learning framework, which distills the 3D human pose structures learned from the fully annotated dataset to in-the-wild images with only 2D pose annotations. Nibali *et al.* [97] proposes a 3D coordinate prediction with flexible statistical modelling capabilities without being memory-intensive, that is differentiable and that spatially generalise well by predicting 2D marginal heatmaps under an augmented soft-argmax scheme.

Multi-person 3D pose estimation: fewer works tackle the problem of multi-person 3D human pose estimation from monocular images. Some of these perform on a top-down manner by firstly identifying bounding boxes likely to contain a person and then proceed with single-person 3d human pose estimation. Among such approaches, Rogez *et al.* [98] classifies the bounding boxes into a set of K-poses. These are then evaluated by a classifier and later refined. This approach produces multiple proposals per subject that need to be accumulated and fused. Zanfir *et al.* [99] combine constraints such as mutual volume exclusion, joint inference and ground plane estimation with a single person model. Moon *et al.* [100] predict

absolute 3D human root localization and a root-relative 3D single-person for each person independently. All these approaches however rely on the accuracy of the people detector and do not scale well when dealing with scenes containing large number of people.

A different type of approaches are instead those that predict multi-person joint locations in a single pass, from which the 3D pose can be inferred even when considering scenarios with strong occlusions. Mehta *et al.* [101] predict 2D and 3D poses for all the subjects in a single forward pass regardless of the number of people in the scene. This approach is robust to occlusions however it heavily relies on 2D estimations, limiting the accuracy to that of the 2D module. Zanfir *et al.* [102] utilize a multitask DNN where the person grouping problem is arranged as an integer program based on learned body part scores parameterized by both 2D and 3D information.

2.2 MULTI-VIEW CONFIGURATION

An example of a more complex system than a monocular configuration, is one relying on multiple input images (with certain viewing angles and overlapping) used for estimating the 3D skeletal representation.
The computation of 3D points in a space by using n-views together with the camera intrinsic and extrinsic is one of the most studied problems in computer vision. These set of publications fall outside the scope of the work presented in thesis as they tackle a subset of different problems, most of which are not of interest for 3D human pose estimation. I will therefore not address this body of work in the related work section. For an overview of such approaches please refer to [103].

Better accuracy can be achieved due to geometrical constraints that can be exploited in the model as well as consistency of information across the multiple views. Initial non-deep learning approaches [104, 105, 106, 107] addressed the problem by optimizing simple parametric models of the human body to match hand-crafted image

features in each view.

As with other research fields, the wide usage of deep learning produced a drastic accuracy improvement in the results. Elhayek *et al.* [108] approach this problem by fusing 2D body part detections, from a ConvNet-based 2D pose estimation, with a generative model-based multi-view tracking algorithm to reconstruct human pose in indoor and outdoor datasets. Tremble [109] made use of a CNN trained on probabilistic visual hull data obtained from multi-viewpoint videos, and an LSTM framework to exploit the temporal continuity of reconstructions. Pavlakos *et al.* [110] introduces a geometry-driven multi-view approach that automatically annotated images with 3D poses starting from generic 2D detections [74]. Their harvested 3D poses are used to demonstrate their effectiveness in two applications: 2D pose penalization and training a ConvNet from scratch for single view 3D human pose prediction. Iskakov *et al.* [111] used a different strategy by projecting heatmaps to a 3D volume using a differentiable model and in a sequential step regresses the estimated root-centered 3D pose through a learnable 3D convolutional neural network. This allows for an end-to-end model training.

Unlike the previously described approaches, Ladkhodamohammadi *et al.* [112] propose a lightweight solution that concatenate 2D detections and pre-process this in a fully connected network to predict global 3d joint positions. Qiu *et al.* [113] instead incorporate multi-view priors in the model. The process is divided in two steps where the first is responsible of estimating 2D joint predictions using multi-view images, whereas the second recovers the 3D pose by using the 2D multi-view predictions of the previous step. Similarly, Remelli *et al.* [114] present a lightweight solution by exploiting 3D geometry to fuse input images into a unified latent representation of pose, which is disentangled from camera view-points. This allows them to reason effectively about 3D pose across different views without using computer intensive volumetric grids.

In this thesis, unlike approaches such as [108, 110, 109, 113], we do not perform pose estimation[1] for each view before fusing them in a final stage. Instead, we generalize multi-stage approaches [15, 16] to multiple views, and iteratively seek an estimate consistent over all views.

2.3 EXTERNAL CAMERA VIEWPOINT

The related work that has been previously described in this section for both monocular and multi-view configurations share one property: the camera set-up is such that the camera/s is/are externally placed in the environment, pointing at the actor or actors that need to be 3D pose reconstructed. We are going to refer to this configuration as *outside-in camera set-up*. Such configuration has been the only possible way of tackling the problem of 3D human pose estimation for a long time. However, the advancement of hardware miniaturization it is not possible to use body-cameras as a mean for body pose reconstruction. This new research task is what is called *Egocentric 3D Human Pose Estimation*.

In this section I am going to introduce a new class of approaches, egocentric 3d pose estimators — also called *first person* 3d pose estimators — where a wearable system such as a head-mounted camera is used to collect the information used to infer the pose of the person.

2.3.1 FIRST PERSON CAMERA VIEWPOINT

While capturing users from an egocentric camera perspective for activity recognition has received significant attention in recent years [115, 116, 117], most methods detect, at most, only upper body motion (hands, arms or torso). Capturing full 3D body motion from head-mounted cameras is considerably more challenging due to the unique perspective effects of an egocentric viewpoint and to the severe self-occlusions. Some head-mounted capture systems are based on RGB-D input and reconstruct mostly hand, arm and torso motions [118, 119]. Jiang and Grauman [120]

[1] Silhouettes in the case of [109].

reconstruct full body pose from footage taken from a camera worn on the chest by estimating ego motion from the observed scene, but their estimates lack accuracy and have high uncertainty.

A step towards dealing with large parts of the body not being observable was proposed in [121] though for external camera viewpoints. Rhodin *et al.* [122] pioneered the first approach towards full-body capture from a helmet-mounted stereo fish-eye camera pair. The cameras were placed around 25 cm away from the user's head, using telescopic sticks, which resulted in a fairly cumbersome setup for the user but with the benefit of capturing large field of view images where most of the body was in view.

Monocular head-mounted systems for full-body pose estimation have more recently been demonstrated by Cha *et al.* [123] and Xu *et al.* [124] (who propose a real-time compact setup mounted on a baseball cap) although in both cases the egocentric camera is placed considerably further from the user's forehead than in our proposed approach and none of them make their code or data available for comparison.

2.3.2 POSE ESTIMATION FROM SENSORS

Inertial Measurement Units (IMUs) worn by the subject provide a camera-free alternative solution to first person human pose estimation. However, such systems are intrusive and complex to calibrate. While reducing the number of sensors leads to a less invasive configuration [125] recovering accurate human pose from sparse sensor readings becomes a more challenging task. An alternative approach, introduced by Shiratori *et al.* [126] consists of a multi-camera structure-from-motion (SFM) approach using 16 limb-mounted cameras. Still very intrusive, this approach suffers from motion blur, automatic white balancing, rolling shutter effects and motion in the scene, making it impractical in realistic scenarios.

CHAPTER 3

POSE FROM MONOCULAR IMAGE

3.1 OVERVIEW

Estimating the full 3D pose of a human from a single RGB image is one of the most challenging problems in computer vision. It involves tackling two inherently ambiguous tasks. First, the 2D location of the human joints, or landmarks, must be found in the image, a problem plagued with ambiguities due to the large variations in visual appearance caused by different camera viewpoints, external and self occlusions or changes in clothing, body shape or illumination. Next, lifting the coordinates of the 2D landmarks into 3D from a single image is still an ill-posed problem – the space of possible 3D poses consistent with the 2D landmark locations of a human, even with perfect 2D landmark locations, is infinite. Finding the physically valid, and anatomically correct 3D pose that matches the image requires injecting additional information usually in the form of 3D geometric pose priors and temporal or structural constraints.

Most research in human pose estimation from images focuses on solving one of these tasks while ignoring the other, *i.e.* estimating the 2D image coordinates of landmark locations given a single RGB image, or solving for 3D pose given known 2D landmark positions as input. However, decoupling these interrelated problems comes at a price. 2D pose recovery algorithms do not leverage 3D geometric information that could aid localization, 3D human pose estimation typically assumes

perfect landmark detections and ignores uncertainty in the location estimates.

We propose a new joint approach to 2D landmark detection and full 3D pose estimation from a single RGB image that takes advantage of reasoning jointly about the estimation of 2D and 3D landmark locations to improve both tasks. We propose a novel CNN architecture that learns to combine the image appearance based predictions provided by *convolutional-pose-machine* style 2D landmark detectors [15], with the geometric 3D skeletal information encoded in our novel pretrained model of 3D human pose. Our probabilistic model of 3D human poses is learned exclusively from 3D mocap data and encodes the space of valid human 3D.

The information captured by our 3D human pose model is embedded in the end-to-end CNN architecture as an additional layer that lifts 2D landmark coordinates into 3D while imposing that they lie on the space of physically plausible poses. The advantage of integrating the output proposed by the 2D landmark location predictors — based purely on image appearance — with the 3D pose predicted by a probabilistic model, is that the 2D landmark location estimates are improved by guaranteeing that they satisfy the anatomical 3D constraints encapsulated in the human 3D pose model. In this way, both tasks clearly benefit from each other.

Many earlier approaches to reasoning jointly about 2D joint estimation and 3D pose reconstruction focused on designing algorithms that can incorporate the inaccuracies of 2D joint detections into the 3D reconstruction task without attempting to re-estimate the 2D joint locations using 3D information [86, 85].

We take this a step further and design a multistage CNN trained to estimate the 2D joint locations. Crucially our new deep architecture includes a novel layer, based on a pre-learned 3D pose model, that injects 3D structural information into the 2D joint estimation. 3D information about the skeletal structure encoded in this layer is propagated to the 2D convolutional layers. In this way, the learning is end-to-end

and the prediction of 2D pose benefits from the 3D information encoded.

A further advantage of our approach is that our 2D and 3D training data sources may be completely independent. Our deep architecture only needs that images annotated with 2D poses, not 3D poses. The human pose model is trained independently and exclusively from 3D mocap data. This decoupling between 2D and 3D training data presents a huge advantage since we can augment our training sets completely independently. For instance we can take advantage of extra 2D pose annotations without the need for 3D ground truth or extend the 3D training data to further mocap datasets without the need for synchronized 2D images.

Our contribution: In this chapter, we show how to integrate a pre-learned 3D human pose model directly within a novel architecture CNN for joint 2D landmark and 3D human pose estimation. In contrast to pre-existing methods, we do not take a pipeline approach that takes 2D landmarks as given. Instead, we show how such a model can be used as part of the CNN architecture itself, and how the architecture can learn to use physically plausible 3D reconstructions in its search for better 2D landmark locations. Furthermore, due to the nature of our architecture, a dataset with both 2D and 3D annotation is not required to train our model, but rather two distinct datasets, one with 2D information and one with 3D data. This allows us to exploit the amount of data contained in the well know datasets for each of the two tasks, with the possibility of increasing their size by merging other datasets. Our method achieves state-of-the-art results on the Human3.6M dataset both in terms of 2D and 3D errors.

3.2 3D POSE DETECTION FRAMEWORK

Our proposed framework to tackle the problem of 3D pose detection from a monocular RGB camera, is a hybrid method in contrast to the previously mentioned pipeline approaches (see Sec. 2.1), since it exploits all the available information in all the modules of the architecture.

Figure 3.1 illustrates the main contribution of our approach, a new multi-stage CNN architecture that can be trained end-to-end to estimate jointly 2D and 3D joint locations. Crucially it includes a novel layer, based on our probabilistic 3D model of human, responsible for lifting 2D poses into 3D and propagating 3D information about the skeletal structure to the 2D convolutional layers. In this way, the prediction of 2D pose benefits from the 3D information encoded. Each stage t produces as output a set of belief maps (heatmaps) for the location of the 2D landmarks (one per joint). The heatmaps from each stage t, combined with learned image features, are used as input to the next stage $t+1$ (grey arrows in the figure show information flowing from one stage to the next). Finally, the last operation generates the most accurate 3D pose.

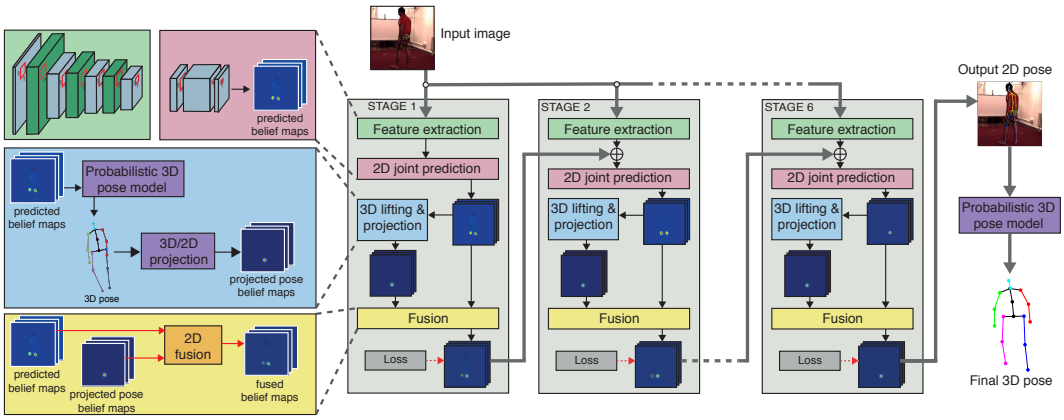


Figure 3.1: Hybrid architecture for monocular pose detection

Section 3.2.1 describes the proposed probabilistic 3D model of human pose, trained on a dataset of 3D mocap data. Section 3.2.2 describes all the new components and layers of the CNN architecture. Finally, Section 3.4 describes the experimental

evaluation of the approach with quantitative results on the Human3.6M dataset, as well as some qualitative results on images from the MPII [127] and LEEDS datasets [128].

3.2.1 PROBABILISTIC MODEL

One fundamental challenge in creating models of human poses lies in the lack of access to 3D data of sufficient variety to characterize the space of human poses. To compensate for this lack of data we identify and eliminate confounding factors such as rotation in the ground plane, limb length, and left-right symmetry that lead to conceptually similar poses being unrecognized in the training data.

We eliminate some factors by simple pre-possessing. Variance due to limb-length is addressed by normalizing the data such that the sum of squared limb lengths on the human skeleton is one; while left-right symmetry is exploited by flipping each pose in the x-axis and re-annotating left as right and vice-versa.

3.2.1.1 Aligning 3D Human Poses in the Training Set

Allowing for rotational invariance in the ground-plane is more challenging and requires integration with our data model. We seek the optimal rotations for each pose such that after rotating the poses they are closely approximated by a low-rank compact Gaussian distribution.

We formulate this as a problem of optimization over a set of variables. Given a set of N training 3D poses, each represented as a $(3 \times L)$ matrix $\mathbf{P_i}$ of 3D joint locations, where $i \in \{1,2,..,N\}$ and L is the number of human joints; we seek global estimates of an average 3D pose μ, a set of J orthonormal basis matrices[1] $\mathbf{e}$ and noise variance σ, alongside per sample rotations R_i and basis coefficients a_i such that the following estimate is minimized

[1] When we say $\mathbf{e}$ is a set of orthonormal basis matrices we mean that each matrix, if unwrapped into a vector, is of unit norm and orthogonal to all other unwrapped matrices.

$$\arg\min_{\mathbf{R},\mu,a,\mathbf{e},\sigma} \sum_{i=1}^{N} \Big(||\mathbf{P_i} - \mathbf{R_i}(\mu + a_i \cdot \mathbf{e})||_2^2 + \sum_{j=1}^{J} (a_{i,j} \cdot \sigma_j)^2 + \ln \sum_{j=1}^{J} \sigma_j^2 \Big) \tag{3.1}$$

Where $a_i \cdot \mathbf{e} = \sum_j a_{i,j}\mathbf{e_j}$ is the tensor analog of a multiplication between a vector and a matrix, and $||\cdot||_2^2$ is the squared Frobenius norm of the matrix. Here the y-axis is assumed to point up, and the rotation matrices R_i considered are ground plane rotations of the form

$$R_i = \begin{pmatrix} \cos\theta_i & 0 & \sin\theta_i \\ 0 & 1 & 0 \\ -\sin\theta_i & 0 & \cos\theta_i \end{pmatrix} \tag{3.2}$$

With the large number of 3D pose samples considered (of the order of 1 million when training on the Human3.6M dataset [129]), and the complex inter-dependencies between samples for $\mathbf{e}$ and σ, the memory requirements means that it is not possible to solve directly as a joint optimization over all variables using a non-linear solver such as Ceres [130]. Instead, we carefully initialize and then alternate between performing closed-form PPCA [131] to update estimates of $\mu, a, \mathbf{e}, \sigma$; and updating the rotations R_i using Ceres [130] to minimize the above error.

From this, we find the planar rotations that minimize the distance between each sample and the rest shape, and alternate between making estimates of μ, a, and $\mathbf{e}$ using probabilistic PCA, and re-optimizing the rotation. As we do this, we steadily increase the size of the basis from 1 through to its target size J. This stops apparent deformations that could be resolved through rotations from becoming locked into the basis at an early stage, and empirically leads to lower cost solutions.

To initialize we use a variant of the Tomasi-Kanade [132] algorithm to estimate the mean 3D pose μ. As the y component is not altered by planar rotations, we take as our estimate of the y component of μ, the mean of each point in the y direction. For

the x and z components, we interleave the x and z components of each sample and concatenate them into a large $2N \times L$ matrix $\mathbf{M}$, and find the rank two approximation of this such that $\mathbf{M} \approx \mathbf{A} \cdot \mathbf{B}$. We then calculate $\hat{\mathbf{A}}$ by replacing each adjacent pair of rows of $\mathbf{A}$ with the closest orthonormal matrix of rank two, and take $\hat{\mathbf{A}}^{\dagger}\mathbf{M}$ as our estimate[2] of the x and z components of μ.

The end result of this optimization is a compact low-rank approximation of the data in which all reconstructed poses appear to have the same orientation. This is particularly clear by looking at Fig. 3.2, where it is possible to see how *standing-up* poses a), b), c) and d) are all close to each other and far from *sitting-down* poses f) and h) which form another clear cluster.

In the next section we extend our model to be described as a multi-modal distribution to better capture the variations in the space of 3D human poses.

3.2.1.2 A Multi-Modal Model of 3D Human Pose

Although it is possible to directly use the learned Gaussian model in the previous section 3.2.1.1 to estimate the 3D (see results in section 3.4), inspection of figure 3.2 shows that the data cannot be represented as Gaussian distribution and is better described using a multi-modal distribution. In doing this, we are heavily inspired both by approaches such as [133] which characterize the space of human poses as a mixture of PCA bases, and by related works such as [134, 52] that represent poses as an interpolation between exemplars. These approaches are extremely good at modeling tightly distributed poses (e.g. walking) where samples in the testing data are likely to be close to poses seen in training. This is emphatically not the case in much of the Human3.6M dataset, which we use for evaluation. Zooming in on the edges of Figure 3.2 reveals many isolated paths where motions occur once and then are never revisited again.

Nonetheless, it is precisely these regions of low-density that we are interested in

[2] $\mathbf{A}^{\dagger}$ being the pseudo-inverse of $\mathbf{A}$.

modeling. As such, we seek a coarse representation of the pose space that says something about the regions of low density but also characterizes the multi-modal nature of the pose space. Therefore, we choose to represent our data as a mixture of probabilistic PCA models using few clusters, and trained using the EM-algorithm [131]. When using a small number of clusters, it is important to initialize the algorithm correctly, as accidentally initializing with multiple clusters about a single mode, can lead to poor density estimates. To initialize our clusters we make use of a simple heuristic.

We first sub-sample the aligned poses (which we refer to as P), and then compute the Euclidean distance d among pairs. We seek a set of k samples S such that the

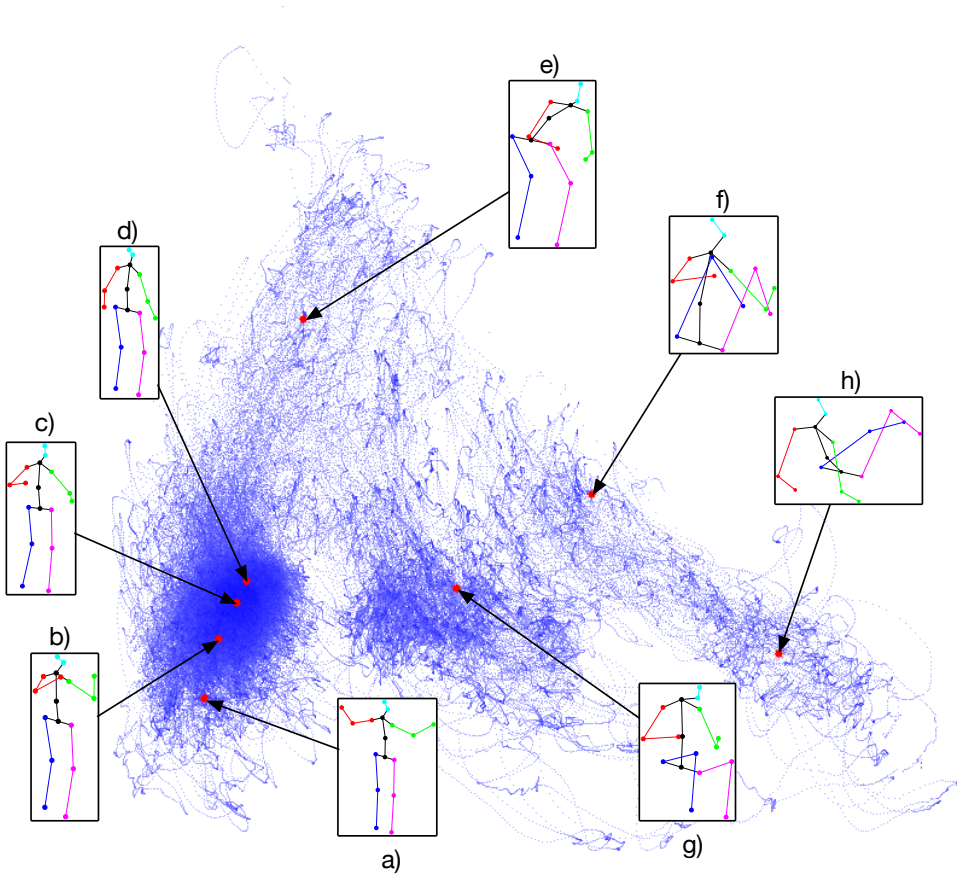


Figure 3.2: Pose alignment of 3D poses. *a-h* represent the 3D poses corresponding the red points highlighted in the higher dimensional pose-space, whereas blue are all the points defining the space.

distance between points and their nearest sample is minimized

$$\arg\min_{S} \sum_{p \in P} \min_{s \in S} d(s, p) \tag{3.3}$$

We find S using greedy selection: we iteratively hold our previous estimate of S constant and select the next candidate s such that $\{s\} \cup S$ minimizes Eq. 3.3. A selection of 3D pose samples found using this procedure can be seen in the rendered poses of Figure 3.2. In practice, we stop proposing candidates when they occur too close to the existing candidates, as shown by samples (a–d), and only choose one candidate from the dominant mode. This prevents the algorithm from clustering too much denser areas.

Given these candidates for cluster centers, we assign each aligned point to a cluster representing its nearest candidate and then run the EM algorithm of [131], building a mixture of probabilistic PCA bases.

3.2.2 2D TO 3D POSE INFERENCE

Our 3D pose inference from a single RGB image makes use of a multistage deep convolutional architecture, that repeatedly fuses and refines 2D and 3D poses, and a second module which takes the final predicted 2D joint positions and lifts them one last time into 3D space for our final estimate (see Figure 3.1).

At its heart, our architecture is a novel refinement of the Convolutional Pose Machine of Wei *et al.* [15], who reasoned exclusively in 2D, and proposed an architecture that iteratively refined 2D pose estimations of joints using a mixture of knowledge of the image and of the estimates of joint locations of the previous stage. We modify this architecture by generating, at each stage, projected 3D pose belief maps which are fused in a learned manner with the standard maps. From an implementation point of view this is done by introducing two distinct layers, the *probabilistic 3D pose layer* and the *fusion layer* (see figure 3.1).

Figure 3.3 shows how the 2D uncertainty in the belief maps is reduced at each stage

of the architecture and how the accuracy of the 3D poses increases throughout the stages: *Top*) Evolution of the 2D skeleton after projecting the 3D points back into the 2D space; *Center*) Evolution of the beliefs for the joint positions *Left hand* through the stages; *Bottom*) 3D skeleton with the relative mean error per joint in millimeters. Even when the estimated locations are incorrect, the model returns a physically plausible solution.

3.2.2.1 Architecture of each stage

The entire architecture consists of 6 stages. Internally, each stage learns to combine *a)* appearance-based belief maps provided by convolutional 2D joint predictors, with *b)* projected pose belief maps, proposed by our new probabilistic 3D pose model that encodes 3D structural information.

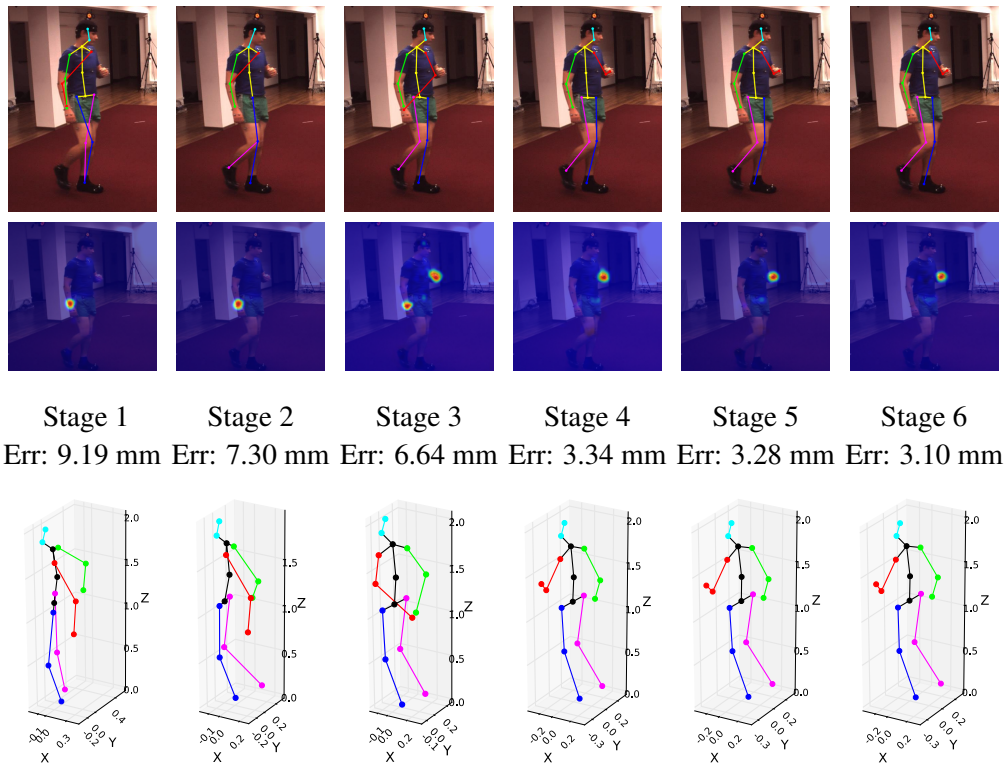


Figure 3.3: Estimation of the 2D and 3D skeleton throughout the stages, showing both a 2D and 3D accuracy improvement as we move towards the output. Specifically, every stage performs a refinement operation that fixes some miss predictions or improves the results.

The accuracy of the 2D and 3D joint locations increases progressively through the stages. Each stage in our sequential architecture is made out of 4 clear components (see figure 3.1).

Predicting CNN-based belief-maps: we use a set of convolutional and pooling layers, equivalent to those used in the original CPM architecture [15], that combine evidence obtained from image learned features with the belief maps obtained from the previous stage $(t-1)$ to predict an updated set of belief maps for the 2D human joint positions. This step is described in section 3.2.2.2.

Lifting 2D belief-maps into 3D: the output of the CNN-based belief maps is then input to a new layer that uses our new pre-trained *probabilistic 3D human pose model* to lift the proposed 2D poses into 3D. The process of lifting 2D into 3D is described in section 3.2.2.3.

Projected 2D pose belief maps: The 3D pose estimated by our 3D pose inference layer is then projected back onto the image plane to produce a new set of projected pose belief maps. These belief maps encapsulate 3D dependencies between the body parts. We describe the projection from 3D to 2D in section 3.2.2.4.

2D Fusion layer: The final layer in each stage (described in section 3.2.2.5) learns the weights to fuse the two sets of belief maps into a unique set which is then input into the next stage $t+1$ of the architecture. It fuses: *(i)* "appearance-based belief maps", proposed by the 2D joint predictors (a set of convolutional layers) and *(ii)* "3D manifold belief maps", proposed by our new manifold-layer.

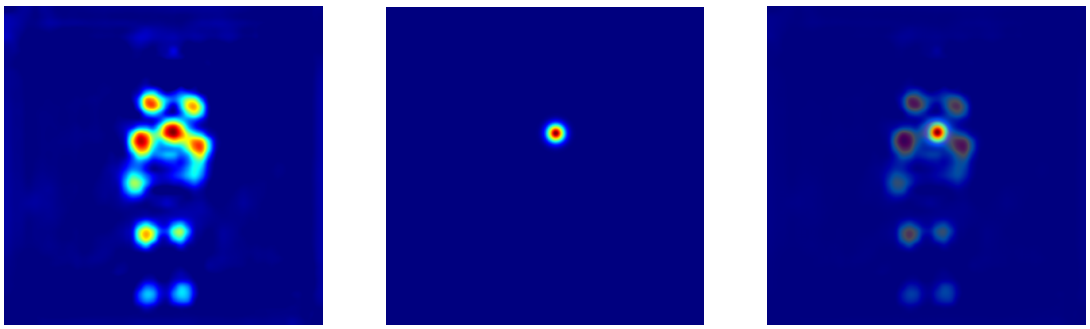
The novel layers were implemented as an extension of the published code of Convolutional Pose Machines [15] inside the *caffe framework* as python layers, and with all weights updated using Stochastic Gradient Descent with momentum.

3.2.2.2 Predicting CNN-based belief-maps

Convolutional Pose Machines [15] can be understood as an updating of the earlier work of Ramakrishna *et al.* [43] to use a deep convolutional architecture. In both approaches, at each stage t and for each landmark p, the algorithm returns dense per pixel belief maps $\mathbf{b}_t^p[u,v]$, which show how confident it is that a joint center or landmark occurs in any given pixel (u,v).

See Figure 3.4 for a visualization of stage 1 belief maps, where the heatmap corresponding to the *Spine* joint goes through the *fusion layer*, which fuses both the *initial convolutional heatmap* (Fig 3.4a), and the *projected pose estimation* (Fig 3.4b), to generate the *fused heatmap* (Fig 3.4c).

For stages $t \in \{2,\ldots,T\}$ the belief maps are a function of not just the information contained in the image but also the information computed by the previous stage.
In the case of convolutional pose machines, and in our work which uses the same architecture, a summary of the convolution widths and architecture design is shown in Figure 3.1, with more details of training given in Convolutional Pose Machines (CPM) [15].
Both approaches [15, 43] predict joints in the image with a different skeletal structure than the one used by the Human3.6M dataset, both in terms of number of joints as well as their connectivity. As such the input and output layers in each stage of



(a) Stage 1 b_p **(b)** Projected $\hat{b}_p$ **(c)** Fusion F^p

Figure 3.4: Evolution of heatmaps through fusion layer

the architecture are replaced with a larger set to account for the greater number of joints. The new architecture is then initialized by using the weights with those found in CPM's model for all pre-existing layers with the new layers randomly initialized. After retraining, convolutional pose machines return per-pixel estimates of landmark locations, while our techniques for 3D estimation (described in the next section) make use of 2D locations. To transform these heatmaps into locations, we select the most confident pixel as the location of each joint

$$Y_p = \arg\max_{(u,v)} b_p[u,v] \tag{3.4}$$

3.2.2.3 Lifting 2D heatmaps into 3D

We follow Zhou *et al.* [86] in assuming a weak perspective model, and first describe the simplest case of estimating the 3D pose of a single frame using a uni-modal Gaussian 3D pose model as described in section 3.2.1. This model is composed of a mean shape μ, a set of basis matrices $\mathbf{e}$ and variances σ^2, and from this we can compute the most probable sample from the model that could give rise to a projected image.

$$\arg\min_{R,a} ||Y - s\Pi ER(\mu + a\cdot\mathbf{e})||_2^2 + ||\sigma\cdot a||_2^2 \tag{3.5}$$

Where Π is the canonical orthographic projection matrix, E a known external camera calibration matrix, and s the estimated per-frame scale. Although, given R this problem is convex in a and s together[3], for unknown rotation matrix R the problem is extremely non-convex, even if a is known and prone to sticking in local minima using first or second order gradient descent. The local optima often lie far apart in pose space and getting stuck in poor optima loads to a significantly worse 3D reconstruction.

We take advantage of the matrix R's restricted form that allows it to be parameterized in terms of a single angle θ (see Eq. (3.2)). Rather than attempting to solve

[3]To see this consider the trivial re-parameterization where we solve for $s\mu + b\cdot\mathbf{e}$ and then let $a = b/s$.

this optimization problem using local gradient based methods we quantize over the space of possible rotations, and for each choice of rotation, we hold this fixed and solve for s and a, before picking the minimum cost solution of any choice of R. With fixed choices of rotation the terms $\Pi ER\mu$ and $\Pi ER\mathbf{e}$ can be computed in advance, and finding the optimal a becomes a simple linear least square problem.

This process is highly efficient and by oversampling the rotations and exhaustively checking in $10,000$ locations we can guarantee that a solution extremely close to the global optima is found. In practice, using 20 samples and refining the rotations and basis coefficients of the best found solution using a non-linear least squares solver obtains the same reconstruction, and we make use of the faster option of checking 80 locations and using the best found solution as our 3D estimate. This puts us close to the global optima and has the same average accuracy as finding the global optima. Moreover, it allows us to upgrade from sparse landmark locations to 3D using a single Gaussian at around 3,000 frames a second using python code on a standard laptop.

To solve our mixture of PPCA models solution, we follow [133] and solve for each Gaussian model independently and select the most probable solution.

3.2.2.4 Projecting 3D poses onto 2D belief maps

Our *projected pose model* is interleaved throughout the architecture (see Figure 3.1). The goal is to correct the beliefs regarding landmark locations at each stage, by fusing extra information about 3D physical plausibility. Given the solution R, s, and a from the previous component, we estimate a physically plausible projected 3D pose as

$$\hat{Y}_p = s\Pi ER(\mu + a \cdot \mathbf{e}) \tag{3.6}$$

which is then embedded in a belief map as

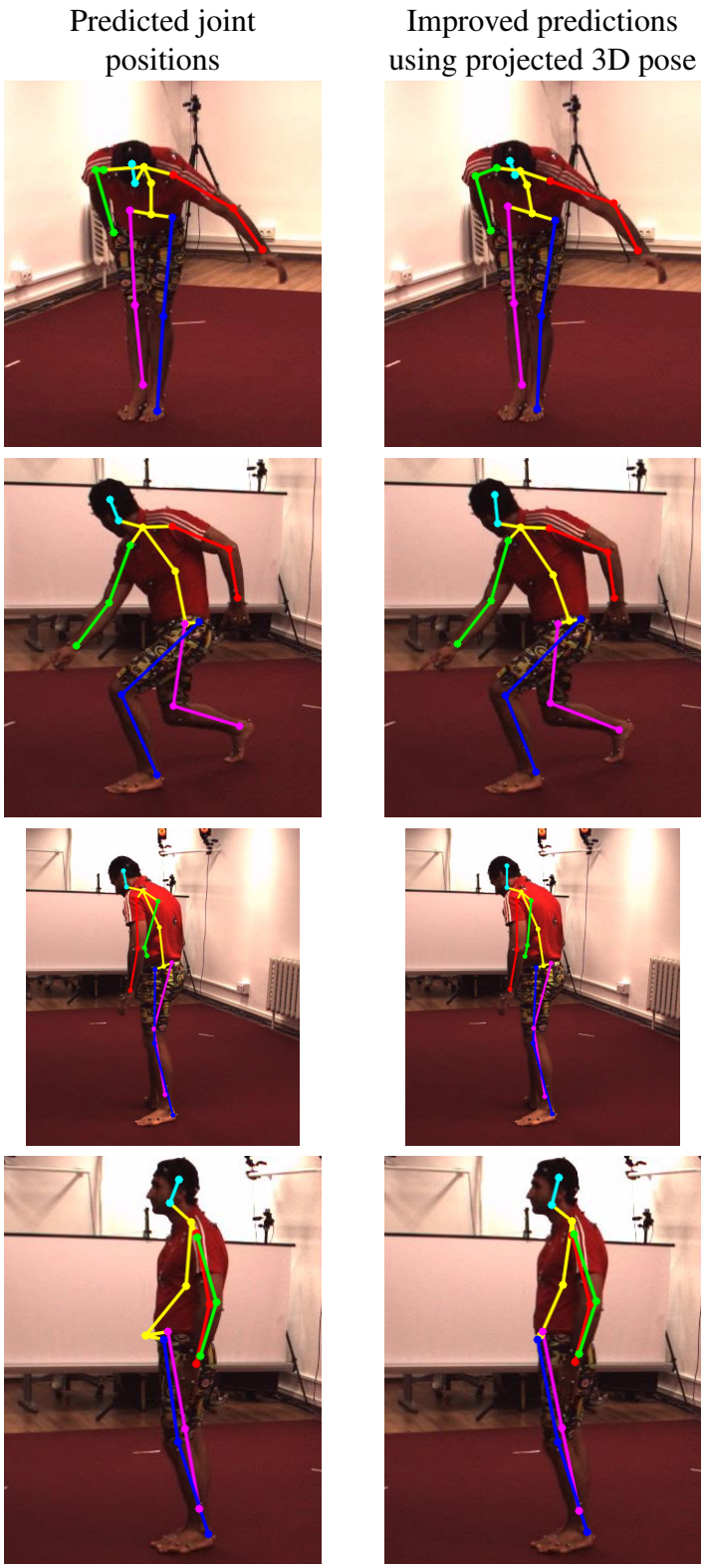


Figure 3.5: Joint prediction refinement

$$\hat{b}^p_{i,j} = \begin{cases} 1 & \text{if}(i,j) = \hat{Y}_p \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

and then convolved using Gaussian filters. The difference in terms of quality of predictions can be seen in Figure 3.5, where results are produced using two models: the first one as a plain CPM architecture, and the second one containing our layers, fusing 2D and 3D information together for better predictions.

3.2.2.5 2D Fusion of belief maps

The 2D belief maps predicted by our *probabilistic 3D pose model* are fused with the CNN-based belief maps b^p according to the following equation

$$f^p_t = w_t * b^p_t + (1 - w_t) * \hat{b}^p_t \tag{3.8}$$

where $w_t \in [0,1]$ is a weight trained as part of the end-to-end learning. This set of fused belief maps f_t is then passed to the next stage and used as an input to guide the 2D re-estimation of joint locations, instead of the belief maps b_t used by convolutional pose machines.

3.2.2.6 The Objective and Training

Following [15], the objective or cost function c_t minimized at each stage is the the squared distance between the generated fusion maps of the layer f^p_t, and ground-truth belief maps b^p_* generated by Gaussian blurring the sparse ground-truth locations of each landmark p.

$$c_t = \sum_{p=1}^{L+1} ||f^p_t - b^p_*||^2_2 \tag{3.9}$$

For end-to-end training the total loss is the sum over all layers $\sum_{t \leq 6} c_t$. Notice that no 3D loss is included here. One could extend this loss to account also for 3D labels, by projecting back the gradient to estimate better pose bases, however this would limit us on having fully annotated datasets.

The novel layers were implemented as an extension of the published code of Convolutional Pose Machines [15] inside the Caffe framework [135] as Python layers, with weights updated using Stochastic Gradient Descent with momentum.
Details of the novel gradient updates used lifting estimates through 3d pose space are given in the appendix A.

3.2.2.7 Final lifting

The belief maps produced as the output of the final stage ($t = 6$) are then lifted into 3D to give the final estimate for the pose (see Figure 3.1) using our algorithm to lift 2D poses into 3D.

3.3 DATASETS

Data driven applications like Deep Learning approaches require a large amount of data to work with. The larger the dataset the better the model can learn how to perform a specific task.

There is a large availability of data on the web for 2D Human Pose Estimation, but there is currently a lack of data in 3D mocap datasets with RGB images. This is also part of the challenges of 3D Human Pose Estimation.

3.3.1 HUMAN3.6M

Our model was trained and tested on the *Human3.6M dataset*, which is one of the largest datasets consisting of 3.6 millions accurate 3D human poses [129], acquired by recording the performance of 5 female and 6 male actors, under 4 different viewpoints (see Figure 4.4[4])

The actions performed by the actors include typical activities like talking on the phone, walking, greeting, eating, etc. (see Fig. 3.7).

[4]Images taken from the Human3.6M dataset's website.

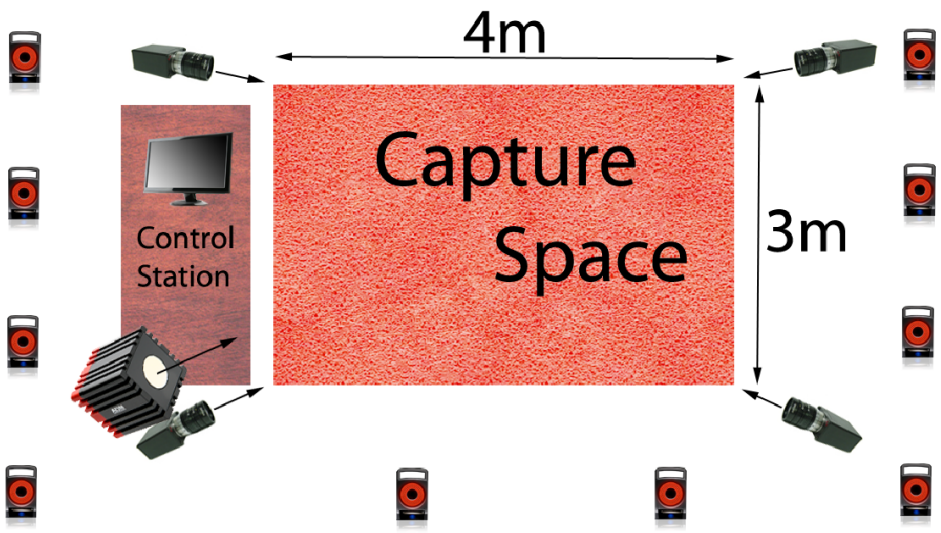


Figure 3.6: Human3.6M camera positions

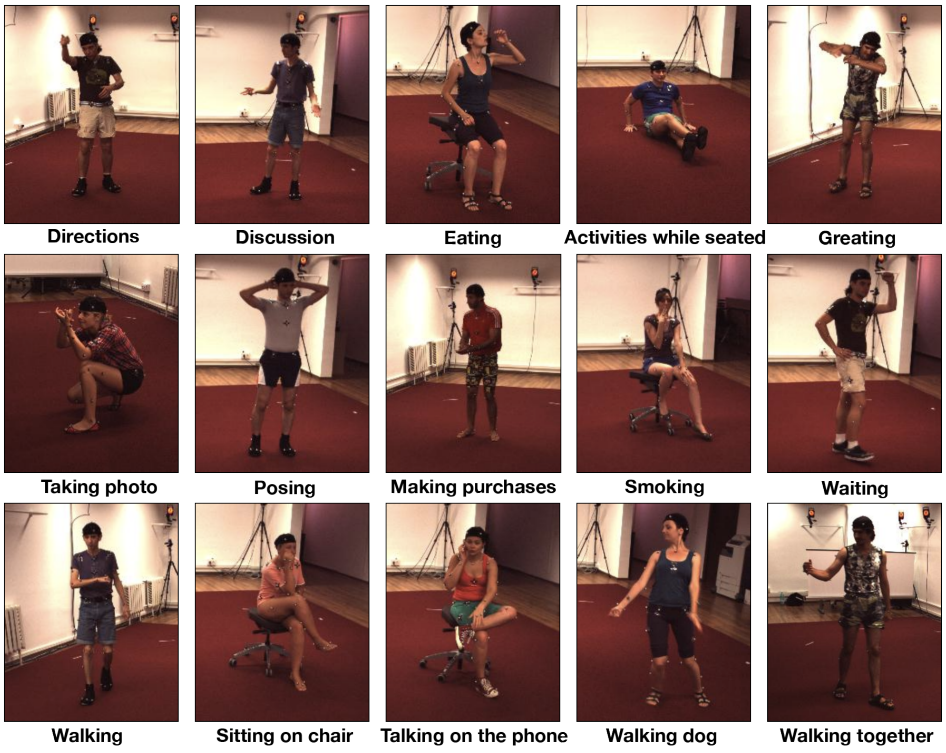


Figure 3.7: Human3.6M actions

During the training of our model, the four different camera views have been considered to be independent. i.e. The same pose captured from different camera views

are considered individually, without exploiting any camera positioning.
The proposed approach has been evaluated both in 2D and 3D on all the different actions, using different evaluation protocols (see Sec. 3.4)

3.3.2 MPII AND LEEDS DATASET

As mentioned in previous sections, one of the main advantages of our *hybrid pipeline approach* consists in the ability to combine different datasets during training, by only using the available information of each individual one. A demonstration of this statement comes from using the MPII [127] and the Leeds [128] datasets in combination with the Human3.6M one.

MPII Human Pose dataset: it includes around 25K images containing over 40K people with annotated body joints, covering a total of 410 human activities, where each image was extracted from a YouTube video.

Leeds Sports Pose Dataset: it contains 2000 pose annotated images of mostly sports people gathered from Flickr. Each image is annotated with 14 joint locations and left and right joints are consistently labeled from a person-centric viewpoint.

Images with 2D annotations are used either from MPII or LEEDS datasets to inform the 2D module of our *hybrid pipeline*, whereas the Human3.6M mocap data are used to learn a 3D lifter that agrees with the 2D estimations provided by the previous module.

Qualitative results on both datasets showing the performance in reconstructions of the proposed approach are shown in Section 3.4.3.

3.4 EXPERIMENTAL EVALUATION

In this section the model is evaluated quantitative on the Human3.6M dataset and additional qualitative results are shown on 2D annotated only datasets like MPII and Leeds.

3.4.1 EVALUATION PROTOCOLS

2D Evaluation: since the goal of the approach is 3D Human Pose Estimation, we limit our evaluation of 2D performance to measuring the pixel distance of predicted joints with their corresponding ground truth location. Therefore, what we refer to as 2D pixel error is defined as:

$$error = \frac{1}{N}\frac{1}{L}\sum_{n}^{N}\sum_{l}^{L}||\hat{P}_l^n - P_l^n||_2$$

where N is the number of frames, L is the number of joints, P_l^n is the ground truth position of joint l for the n-th frame, and $\hat{P}_l^n$ is its predicted location.

3D Evaluation: Several evaluation protocols have been followed by different authors to measure the performance of their 3D pose estimation methods on the Human3.6M dataset.

Protocol 1, the most standard evaluation protocol on Human3.6M was followed by [129, 66, 136, 68, 137, 86, 81]. The training set consists of 5 subjects (S1, S5, S6, S7, S8), while the test set includes 2 subjects (S9, S11). The original frame rate of 50 fps is down-sampled to 10 fps and the evaluation is on sequences coming from all 4 cameras and all trials. The reported error metric is the *3D error* which corresponds to the average Euclidean distance of the estimated 3D joints to the ground truth. The error is averaged over all 17 joints of the Human3.6M skeletal model.

Protocol 2, followed by [138, 139], selects 6 subjects (S1, S5, S6, S7, S8 and S9) for training and subject S11 for testing. The original video is down-sampled to every 64th frame and evaluation is performed on sequences from all 4 cameras and all trials. The error metric reported in this case is the *3D pose error* equivalent to the per-joint 3D error up to a similarity transformation (i.e. each estimated 3D pose is aligned with the ground truth pose, on a per-frame basis, using Procrustes analysis). The error is averaged over a subset of 14 joints.

Protocol 3, followed by [6] selects the same subjects for training and testing as *Protocol 1*. However, evaluation is only on sequences captured from the frontal camera ("cam 3") from trial 1 and the original video is not sub-sampled. The error metric used in this case is the *3D pose error* as described in Protocol 2. The error is averaged over 14 joints

3.4.2 QUANTITATIVE RESULTS

A quantitative evaluation of the proposed approach against other competing approaches is shown in Table 3.1, using *Protocol 1* as the evaluation protocol. Our baseline method using a single unimodal probabilistic PCA model outperforms almost every method in most action types, with the exception of Sanzari *et al.* [81], which it still outperforms on average across the entire dataset. Our mixture model improves on this again, offering a 4.76mm improvement over Sanzari *et al.*, our closest competitor.

Note: some approaches [136, 86] infer the poses by exploiting temporal information, therefore using more information than what has been used by our method, where each frame is processed independently.

Due to the dissimilarity of evaluations by other approaches, we need to perform additional separate comparisons. The reconstruction errors generated using evaluation *Protocol 2* are shown in Table 3.2, where our approach outperforms all other

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases
LinKDE [129]	132.71	183.55	132.37	164.39	162.12	205.94	150.61	171.31
Li *et al.* [66]	-	136.88	96.94	124.74	-	168.68	-	-
Tekin *et al.* [136]	102.39	158.52	87.95	126.83	118.37	185.02	114.69	107.61
Tekin *et al.* [68]	-	129.06	91.43	121.68	-	162.17	-	-
Tekin *et al.* [137]	85.03	108.79	84.38	98.94	119.39	**95.65**	98.49	93.77
Zhou *et al.* [86]	87.36	109.31	87.05	103.16	116.18	143.32	106.88	99.78
Sanzari *et al.* [81]	**48.82**	**56.31**	95.98	**84.78**	96.47	105.58	**66.30**	107.41
Ours - Single PPCA Model	68.55	78.27	77.22	89.05	91.63	110.05	74.92	83.71
Ours - Mixture PPCA Model	64.98	73.47	**76.82**	86.43	**86.28**	110.67	68.93	**74.79**
	Sitting	Sitting Down	Smoking	Waiting	Walk Dog	Walking	Walk Together	Average
LinKDE [129]	151.57	243.03	162.14	170.69	177.13	96.60	127.88	162.14
Li *et al.* [66]	-	-	-	-	132.17	69.97	-	-
Tekin *et al.* [136]	136.15	205.65	118.21	146.66	128.11	65.86	77.21	125.28
Tekin *et al.* [68]	-	-	-	-	130.53	65.75	-	-
Tekin *et al.* [137]	**73.76**	170.4	85.08	116.91	113.72	**62.08**	94.83	100.08
Zhou *et al.* [86]	124.52	199.23	107.42	118.09	114.23	79.39	97.70	113.01
Sanzari *et al.* [81]	116.89	**129.63**	97.84	**65.94**	130.46	92.58	102.21	93.15
Ours - Single PPCA Model	115.94	185.72	88.25	88.73	92.37	76.48	77.95	92.96
Ours - Mixture PPCA Model	110.19	173.91	**84.95**	85.78	**86.26**	71.36	**73.14**	**88.39**

Table 3.1: Evaluation on Human3.6M dataset using Protocol 1 with error expressed in mm.

competitors. Although in this specific case, our model had been trained using only the 5 subjects used for training in *Protocol 1* (one fewer subject), our model still outperforms both other methods [138, 139].

Finally, evaluation *Protocol 3* generates errors shown in Table 3.3. The only other approach using such protocol is Bogo*et al.* [6], where the same sub-set of joints is matched for better comparison.

Our method outperforms Bogo *et al.* [6] by almost 3*mm* on average. It is important to remind the reader that, unlike our approach, Bogo *et al.* [6] exploits a high-quality detailed statistical 3D body model [88] trained on thousands of 3D body scans, that captures both human body shape identities and the body deformation based on the

	Average error
Yasin *et al.* [138]	108.3
Rogez *et al.* [139]	88.1
Ours - Mixture PPCA Model	**70.7**

Table 3.2: Evaluation on Human3.6M dataset using Protocol 2 with error expressed in mm.

	Average error
Bogo *et al.* [6]	82.3
Ours - Mixture PPCA Model	**79.6**

Table 3.3: Evaluation on Human3.6M dataset using Protocol 3 with error expressed in mm.

	2D pixel error
Zhou *et al.* [86][5]	10.85
Trained CPM [15] architecture	10.04
Ours including 3D refinement	**9.47**

Table 3.4: Evaluation of pixel error on Human3.6M dataset

pose the actor is performing. Nonetheless, even with less available data for training, our approach outperforms Bogo *et al.* [6].

Finally, Table 3.4 shows the comparison between our approach and [15, 86] on the 2D prediction error. The 2D error reduction using our full approach over the estimates of [15] is comparable in magnitude to the improvement due to the change of architecture moving from the work Zhou *et al.* [86] to the state-of-the-art 2d architecture [15] (i.e. a reduction of 0.59 pixels vs. 0.81 pixels).

3.4.3 QUALITATIVE RESULTS

Our proposed approach trained exclusively on the Human3.6M dataset can be used to identify 2D and 3D landmarks of images contained in different datasets.

Figure 3.8 shows some qualitative results on the MPII dataset. Our model was not trained on images as diverse as those contained in this dataset, however it often retrieves correct 2D and 3D joint positions. The last row shows example cases where the method fails either in the identification of 2D or 3D landmarks.

Figure 3.9 shows qualitative results on the Leeds dataset, including failure cases.

[5]Results obtained by using temporal smoothness and knowing the action label.

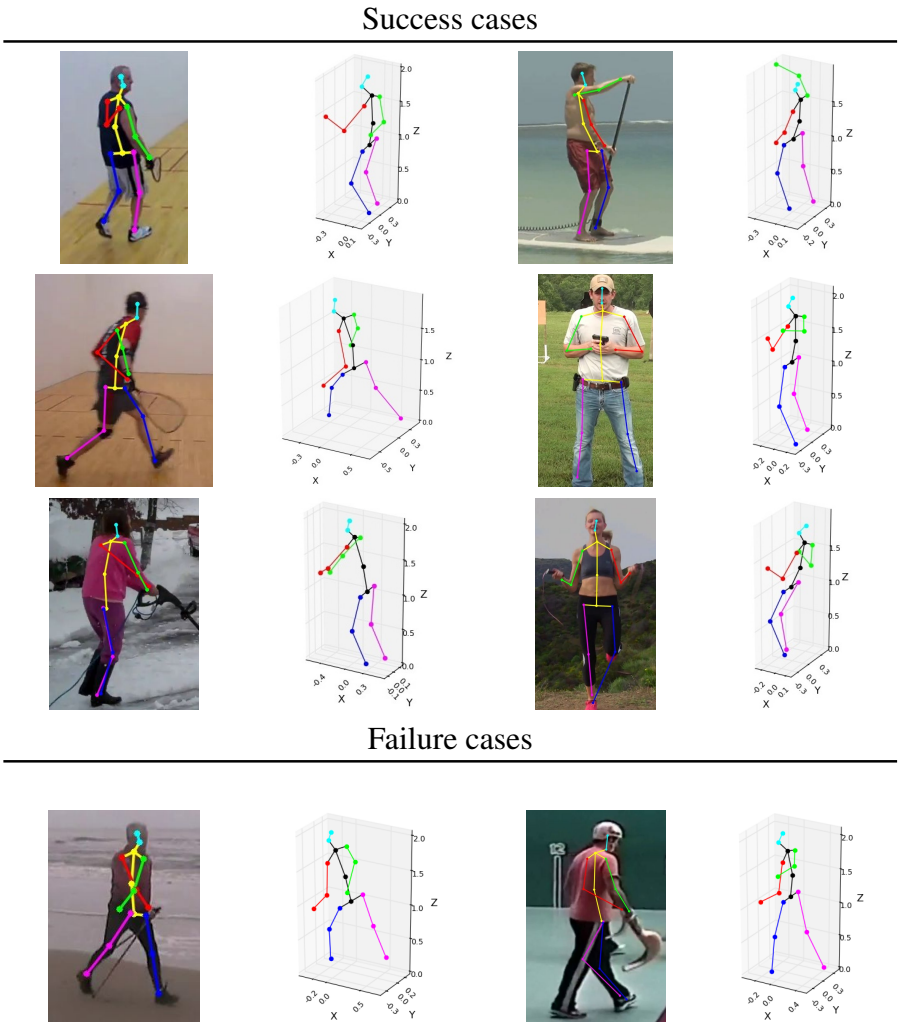


Figure 3.8: Results on images from the MPII dataset. The left failure case is an example in which the 2D pose estimator swaps the arms and the 3D pose module therefore fails. The right failure case instead has the wrong 2D estimation, that is then fixed by the 3D module.

Notice how our *probabilistic 3D pose model* generates anatomically plausible poses even though the 2D landmark estimations are not all correct. However, as shown in on the right, even small errors in 2D pose can lead to drastically different 3D poses. These inaccuracies could be mitigated without further 3D data by annotating additional RGB images for training from different datasets.

Finally, Figure 3.10 show some qualitative results on some sampled frames from the test-set. The identified 2D landmark positions and 3D skeleton is shown for each

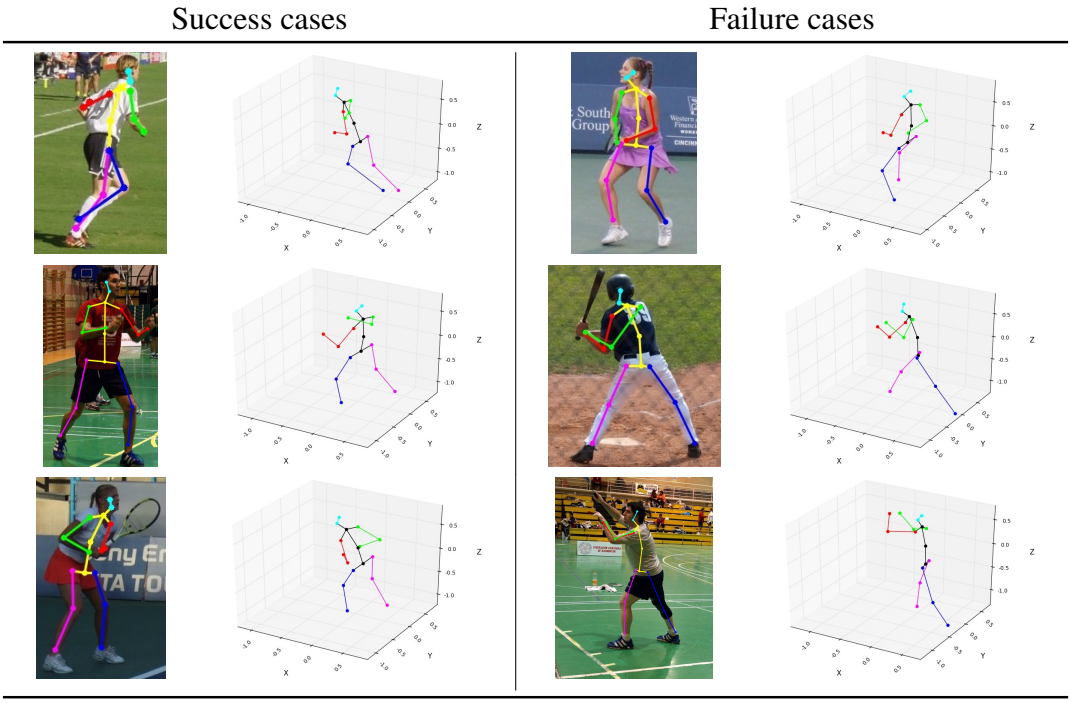


Figure 3.9: Results on images from the Leeds dataset

pose taken from different actions: Walking, Phoning, Greeting, Discussion, Sitting Down.

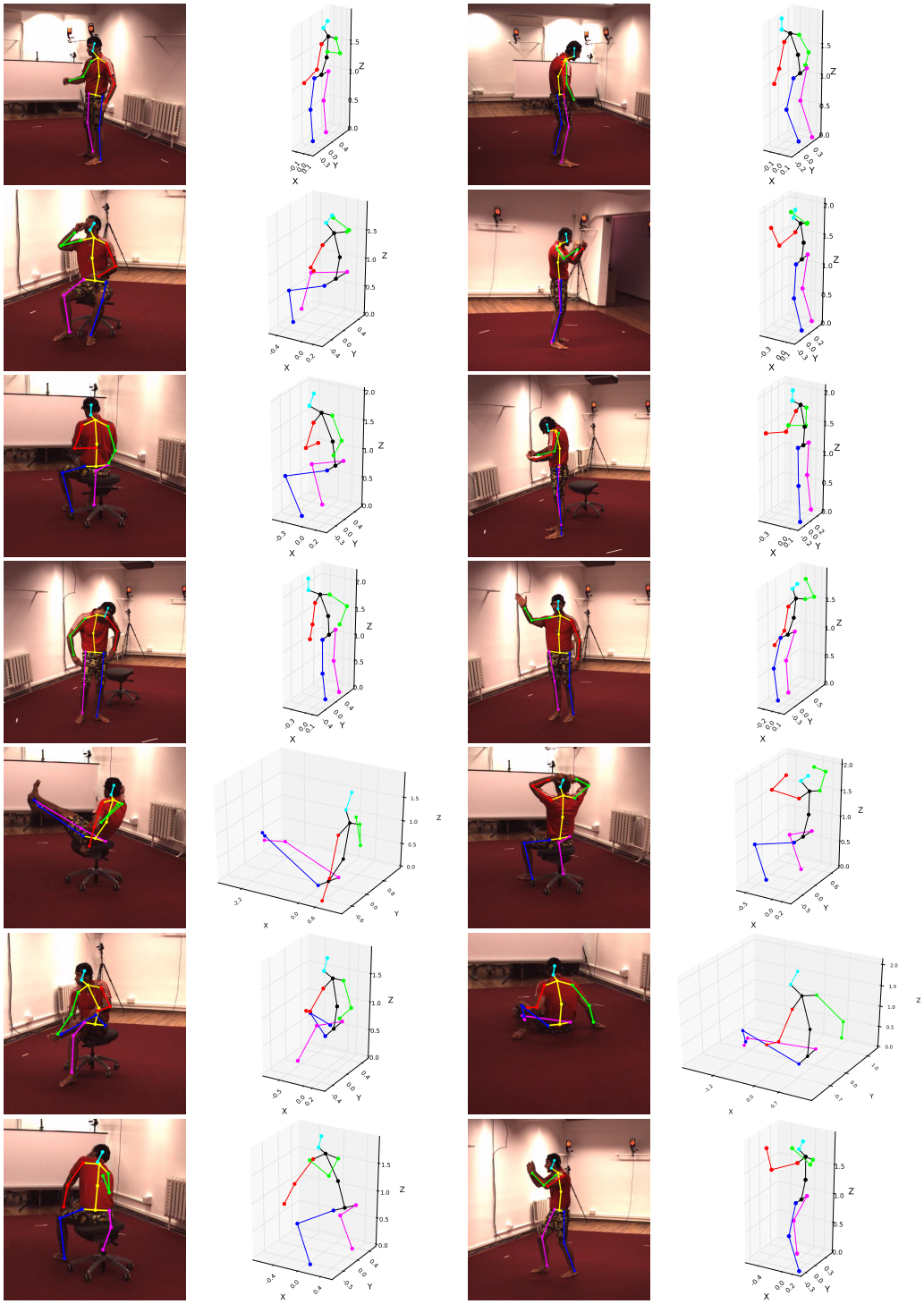


Figure 3.10: Results from the Human3.6M dataset

3.5 CONCLUSION

In this chapter, inspired by the work by [86, 85, 15] we have presented a novel *hybrid pipeline approach* to human 3D pose estimation from a single image that outperforms previous approaches to this problem. To our knowledge we are the first to approach this problem as a problem of iterative refinement in which 3D proposals help refine and improve upon the 2D estimates.

This approach shows the importance of thinking in 3D even for 2D pose estimation within a single image, with the iterative 3D model demonstrating better 2D accuracy than Convolutional Pose Machines [15], the iterative 2D approach it is based upon.

Such approach allows us to estimate the correct 3D pose of a person from an external camera point of view that can be used to guide a robot/robotic arm in the interaction with a person, making sure that person safety is always respected and the task can be completed as designed.

Our novel module for upgrading poses from 2D to 3D is extremely efficient, and runs in CPU-based python at around 1,000 frames a second, while a GPU-based real-time approach for Convolutional Pose Machines has been announced, compared to the current version with 6 stages which runs at 10 fps (current bottle-neck). Intuitively, one could decrease the number of stages to get a faster execution time with a direct drop in accuracy (trade-off between speed and accuracy).
Integrating the faster CPM version with our architecture would provide a reliable real-time 3D pose estimator seems like a natural future direction, as does integrating our approach with a simpler 2D approach for real-time pose estimation on lower power devices.

CHAPTER 4

POSE FROM MULTI-CAMERA VIEWS

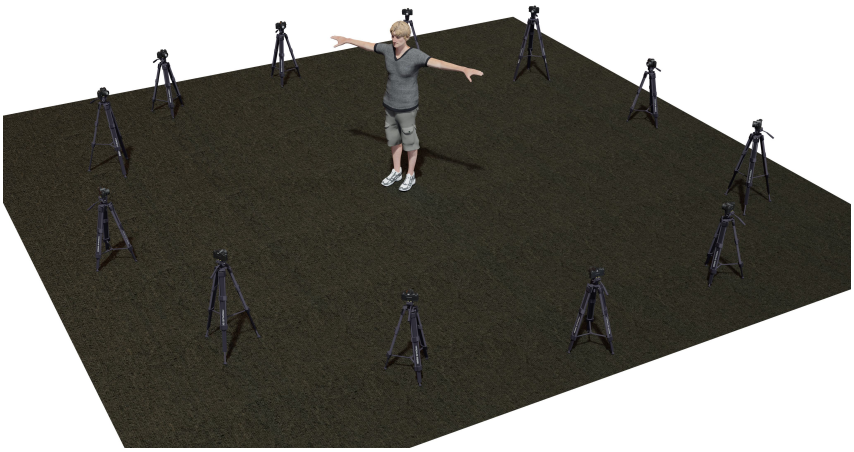


Figure 4.1: Multi-view 3D human pose estimation set-up.

In this chapter a less trivial but more precise configuration is presented: a multi-camera set-up, as shown in Figure 4.1. To goal is to have such approach to be working with images captured in a natural environment, not only in an indoor studio.

4.1 OVERVIEW

One fundamental challenge in the 3D estimation of dynamic and moving objects lies in finding a rich source of ground-truth data. This is not just a problem for modern learning based approaches, that require an abundance of data in order to make

inferences about the world, but also for the traditional ones such as model-based reasoning that make heavy use of constraining prior information about the world. Even these traditional methods rely on carefully tuned parameters which control expressiveness of the model [52], internal connectivity priors [140], or both [141] that must be adjusted to recover plausible reconstructions.

Extracting 3D data from images is a fundamentally ill-posed problem that even people find challenging. Unlike standard image labeling problems, such as Imagenet [142], that make heavy use of human annotation, we cannot simply expect people to reliably annotate images with the distance of joints from the camera. The gold standard for accurately capturing 3D information of full-body human poses data remains using Multi-camera Motion Capture (MoCap) systems. These systems make use of early vision techniques based on the identification of markers across multiple cameras and on the estimation of the 3D location of these points through triangulation.

Such systems generates very reliable annotations, however they also require strong, unambiguous cues to identify the points. In practice, this means that successful MoCap relies on the subject wearing dark tight clothing and brightly colored markers attached to the subject's clothes and their movements captured from multiple cameras, allowing pose reconstruction via triangulation. Even though the quality of these reconstructions is undisputed, there are limitations associated with this procedure. For example, conditions inside studios — such as lighting and backgrounds — must be heavily controlled. The resulting images are not representative of natural images leading to poor generalization images captured in-the-wild. In addition, existing MoCap datasets do not necessarily capture a vast enough set of human poses or sufficient variations across clothes and subjects.

In response to these limitations, some recent works [129, 139, 143] have generated more varied synthetic images using MoCap pose data as the source of the human poses. Although these images are more varied than MoCap data, they are still not

natural images; and these images tend not to capture information and confusion caused by the deformation of loose fitting clothing [144].

Another approach to avoiding these problems is to chain together different regressors based on multiple data sources; one network is trained to predict 2D joint locations in natural images, while a second regressor upgrades these 2D joint locations to 3D using MoCap data. This approach comes with caveats similar to those of the methods discussed above. We might know that a method gives highly accurate 3D poses on MoCap data and good 2D joint locations in natural images, but we remain fundamentally unsure as to its 3D accuracy in natural images.

As such, effective marker-less motion capture is an important tool to train networks to generate reliable 3D models from natural images: not only would it enable the capture of more natural data and reduce the constraints during studio capture, but also to capture outside of the studio and increase the amount of training data without limits.

In response to these difficulties we present a novel architecture, that takes many of the best aspects of the approach introduced in the previous chapter operating on single camera images, and places it in a multi-camera framework, allowing additional sources of data to be exploited.

We present a Huber loss based robust estimator for fusing multi-view 2D pose predictions into a coherent 3D pose, consistent with natural human poses. Unlike existing 3D frameworks, this is not simply done at the end of a pipeline for 2D joint estimation, but is iterated through multiple-stages. This carries substantial benefits. Our use of a robust estimator means that at each stage the 3D model can discard a minority of incorrect 2D joint estimates; the knowledge of where the joints should be in each image is fed back into the algorithm for image-based refinement.

One fundamental question regarding these datasets composed of millions of frames,

such as Human3.6M, is whether they are in fact large enough. The primary issue is whether the dataset is sufficiently diverse to allow trained networks to exhibit good generalization to a held-out test set. Even in restrictive cases, such as the test set used in Human3.6M, where the held-out data consists of new actors performing the same movements in similar clothes in the same studio, there is enough variability in individual body shapes and in how they move that generalization is not guaranteed. To help address this issue, we demonstrate how unlabeled data can be labeled by our algorithm and augment the datasets used for the training of existing methods, leading to overall better performance on standard benchmarks.

We also present a weakly-supervised approach that combines weak supervision from 2D joint labels for which ground truth labels are available, and a self-supervised loss, for an additional corpus of unlabeled images, expressed as the agreement between joints detected in the 2D images and the re-projection of an estimated 3D model. We show how this notion of 2D estimates consistent with a global 3D pose can be exploited for end-to-end self-supervised training.
By using a mixture of weakly labeled (2D labels only) and unlabeled data, we are able to train a network that minimizes the predicted error between 2D estimates and the global 3D position. We show how this form of self-supervision is not restricted to multi-camera setups and can be equally applied to the case of monocular 3D human pose estimation with similar improvements in performance.

We evaluate multiple networks and find consistent multi-millimeter improvement. When the differences between state-of-the-art networks are so small, this raises questions as to whether we are over-fitting and if time would be better spent building larger datasets rather than fighting for small improvements obtained from architectural changes.

Our contribution: in this chapter, we extended existing work on single view reconstruction to a multi-camera setting. We also show how such single view methods can be enhanced by training on multi-view based annotation of unlabeled data.
The use of an iterative, and robust, multistage approach to multi-view reconstruction allows us to correct mistakes in body joint estimations as they arise, and to *think again*, reconsidering the 2D position of joints in the image using interim knowledge of 3D pose. Finally, we show how this can be turned into a weakly-supervised approach.

4.2 MULTI-VIEW FRAMEWORK

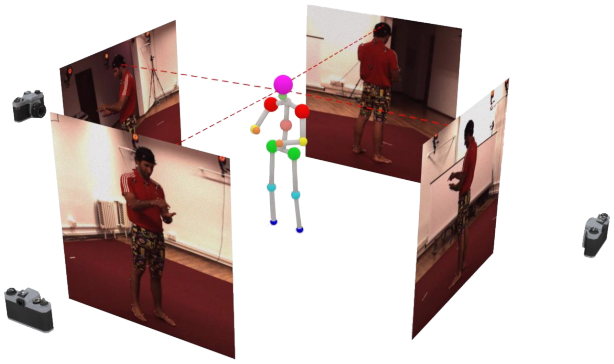


Figure 4.2: Exploiting geometry in multi-view configuration.

Our proposed framework, tackling the problem of 3D pose detection from a multi-view camera setup which differs from the previous chapter due to the possibility to exploit geometrical information as an additional constraint (see Fig. 4.2). The presented work follows the approach described in the previous chapter (see Sec. 3) in maintaining a six-stage Convolutional Neural Network.

Unlike existing approaches such as [108, 110, 109], we do not perform independent

predictions for 2D poses[1] for each view before fusing them in a final stage. Instead, we generalize the multi-stage approach to human pose estimation used by methods such as [15, 16] to multiple views.

Each stage of the CNN (see Fig. 4.3) takes as inputs *a)* the set of images from different cameras we are trying to reconstruct from, and *b)* the set of 2D pose heatmaps predicted in the previous stage for each multi-view image.
Inside each stage the algorithm independently improves the 2D locations of joints in each image and uses them to reconstruct a 3D model consistent with the 2D joint predictions for all the views. Maintaining this internal representation of pose as a 3D model, coherent with all views, allows us to inject 3D information into the learning process. In addition, by re-projecting the 3D model into all the camera views using known camera geometry we can use 2D losses throughout all the stages bypassing the need for 3D annotations associated with the images.

This novel multi-view and multi-stage reconstruction allows us to *rethink joint locations* in light of knowledge of an interim 3D reconstruction, to recover from mistakes made, and to try again to find support in the image for the predictions of joint locations made by a coherent working hypothesis of 3D positions. Details are given in section 4.2.1.

Importantly, our approach maintains the computable sub-gradients used in the previous chapter, described in Appendix A, when generating and projecting the 3D model. This allows the system to be trained end-to-end.
We make substantial changes that improve the robustness of the system while preserving the guarantees of [16] that the model fitting procedure will not get stuck in poor fitting local optima. This is done by replacing the Least Squares procedure (see Sec. 3.2.2.3), with an Iterative Re-weighted Least Squares (IRLS) approach that mimics the Huber loss and preserves convexity for any particular choice of planar

[1] Silhouettes in the case of [109]

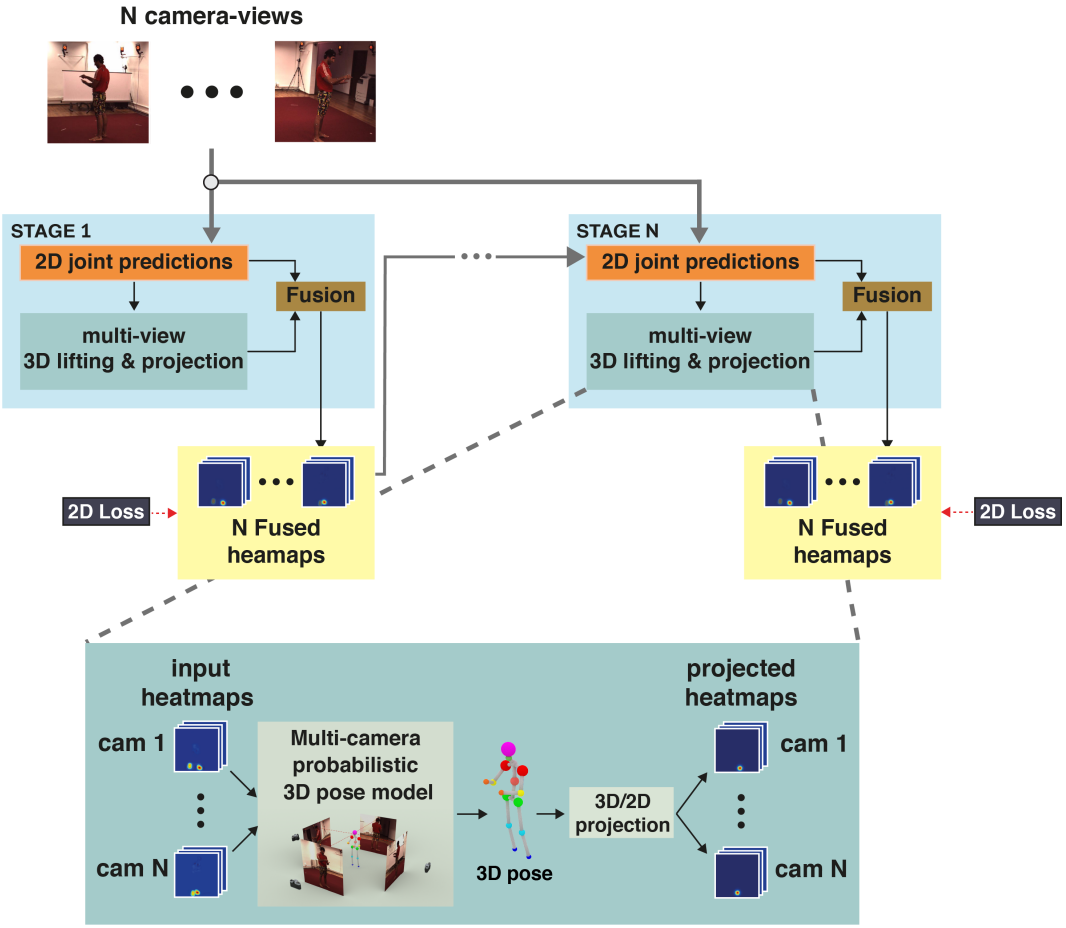


Figure 4.3: Detailed description of the multi-stage architecture designed 3D human pose estimation. The multiple stages serve as a refinement process and all stages following the first one are given as input the predictions estimated by the previous stage. The multi-camera probabilistic model is the key in injecting 3D information back to the the 2D module.

rotation. Details of this are given in section 4.2.1.

4.2.1 ARCHITECTURE

The proposed architecture is a multi-stage convolutional neural network inspired by the work described on Chapter 3, which was in turn an extension of the architecture introduced by Wei *et al.* [15]. They introduced Convolutional Pose Machines (CPM), a multi-stage 2D pose estimator in which each stage performed a refinement of the estimate computed by the previous stage.

As shown in Figure 4.3, the first step in each stage independently predicts, in every camera view, the 2D pose of the person in the image. These predictions take the form of heatmaps generated via a convolutional architecture with the weights shared between all camera views.

These heatmaps are generated by: *a)* a set of convolutional layers *shared* by all stages that are performing feature extraction; followed by *b)* a set of convolutional layers, *unique* to each stage, that compute a heat map representing the location of each joint. All stages (except stage 1) also take as input the heatmaps generated in the previous stage.

The size and connections of these convolutional layers remain the same as in CPM[15]. However, we additionally apply batch normalization before the *ReLu* activation function.

The next step within each stage takes heat-maps as input and computes the 3D pose most consistent with the 2D information provided by each camera view. Heat-maps are then converted into 2D locations by selecting the most confident pixel as the location of each of the joints

$$I_p^c = \arg\max_{(u,v)} H_p^c[u,v]$$

where H_p^c is the heat-map representing joint p for camera view c.

The 2D poses are then used by the multi-camera probabilistic 3D pose estimator (described in section 4.2.1) to generate a single 3D pose that agrees over all the different camera 2D poses. This pose is projected back onto the 2D image for each camera view using a weak perspective projection, and the new projected 2D poses are converted into heat-maps by a Gaussian convolution

$$\hat{H}_p^c[u,v] = \begin{cases} 1 & \text{if}(u,v) = \hat{I}_p^c \\ 0 & \text{otherwise.} \end{cases}$$

where $\hat{I}_p^c$ is joint p of the projected 2D pose in camera c.

The final operation fuses the heat-maps regressed by the convolutional layers with those estimated by projecting the 3D pose into 2D. This fusion is implemented by applying a convolutional layer with filters of size $[1 \times 1]$ and number_joints filters, to each camera view independently, giving a set of heatmaps, one for each choice of joint and camera.

As an implementation detail, all the computations performed on each camera view make use of the same convolutional operations; this enables us to have an efficient implementation by setting the batch size to be equal to the number of cameras and ordering the images appropriately.

4.2.2 3D POSE ESTIMATION

In Chapter 3, we suggested approaching human pose estimation using a formulation inspired by non-rigid structure from motion. Assuming a known basis of human poses given by a set of matrices $\mathbf{e}$, a standard deviations σ and a rest shape μ, we suggested estimating the cost of a particular parameterized human pose, given 2D locations I, as:

$$\underset{s,a,R}{\arg\min} ||I - s\Pi ER(\mu + a \cdot \mathbf{e})||_2^2 + \sigma^2 \cdot a^2 \tag{4.1}$$

Where Π is the canonical orthographic projection matrix, E a known transformation from the world coordinates to those of the camera, R is a planar rotation matrix that describes the rotation of the human pose in the ground-plane, and s is the estimated per-frame scale. Here a is a vector of basis coefficients, $\mathbf{e}$ a 3D tensor

$\mathbf{e} \in \mathbb{R}^{Bases \times Points \times 3}$.

The tensor product $a \cdot \mathbf{e}$ is defined as $\sum_i a_i \mathbf{e}_i$, and the square terms in the final expression refer to an element-wise square. The closest parameterized pose for 2D data I was given by minimizing Eq. 4.1, which can be expressed more compact as

$$\arg\min_{s,a,R} P(s,a,R|I) \tag{4.2}$$

We observed that, for any given choice of rotation, the global minima could be interpreted as an unconstrained linear least squares problem and solved efficiently. Furthermore, we suggested brute forcing over a small set of ground plane rotations to quickly find a global minima without needing to worry about getting stuck in poor quality local optima.

We provide several additions to the framework: *Rotation marginalization* has proven to improve the stability of the model; the introduction of *Principled shape warping* for multiple views, and the a new *robust loss* for outlier rejection which is particularly important when dealing with multi-view systems, since the model needs to be able to disregard proposals that disagree with the predictions from the cameras majority.

4.2.2.1 Rotation marginalization for improved stability

We observed that using more than 80 sampled rotations did not improve the overall accuracy of the reconstructions. Although this is true, the model yields flickering and unstable reconstructions when run on videos. Much of this flicker can be attributed towards trying to reconstruct ambiguous poses that can be equally well explained by two or more different rotations.

We write the optimal reconstruction, given a choice of rotation R as

$$Q_R = Rs(\mu + a \cdot \mathbf{e})$$

where a and s are found by solving the following optimization problem

$$\{s,a\} = \arg\min_{s,a} P(s,a,R|I) \tag{4.3}$$

Marginalizing over the discrete set of rotations $\mathscr{R}$, gives the following 3D body pose estimate

$$\frac{\sum_{R\in\mathscr{R}} \exp(-\rho P(s_R,a_R,R|I))Q_R}{\sum_{R\in\mathscr{R}} \exp(-\rho P(s_R,a_R,R|I))} \tag{4.4}$$

with ρ an arbitrary number which defines the shape of the Gaussian function for the weight and P being the residual of the reconstructed pose using rotation R. This means that instead of only selecting the best rotation for a pose (as done in the previous chapter 3), we are now taking all possible choices of in-plane rotation $\frac{2\pi}{100}n$ with $n \in [0,99]$, for each of those R we find the optimal $3D$ pose, and then a weighted average of all those poses is performed to identify the final 3D pose. Since the exponential weights depend on the residual of each pose Q_R, only the rotations that best explain the $2D$ detections are significantly contributing to the average.

This elimination of flickering is highly desirable, not just in that it makes the reconstructions of video appear more lifelike and appealing to humans, but also in that the stability of the reconstructions carries important semantic information. If we are to use 3D reconstructions of people as a first step in action analysis, the stability and dynamics of the reconstructions contains important information that informs our understanding of the actions.

4.2.2.2 Principled shape warping for multiple views

We approached the problem of reconstruction through the lens of probabilistic PCA [131] with a known basis. After generating a reconstruction from basis co-

efficients, a final stage was to warp the reconstruction to lie closer to the input data. In the context of 3D reconstruction from an single orthographic camera this can be done as post processing, where a weighted average of the x and y coefficients of the image and the reconstruction Q_R are taken together while the z component remains constant.

When multiple cameras are being used, this fusion between the model and the data can not be performed as a simple post-processing step. Instead, we jointly estimate a new shape $\tilde{Q}$ consistent with all frames and close to the model estimate.
Given a rotation R, this can be written as

$$\underset{\tilde{Q}_R,s,a}{\arg\min}\,\lambda \sum_{c\in\mathscr{C}} ||I_C - \Pi E \tilde{Q}_R||_2^2 + ||\tilde{Q}_R - sR(\mu + a\cdot\mathbf{e})||_2^2 + \sigma^2\cdot a^2 \qquad (4.5)$$

where $\mathscr{C}$ refers to a set of cameras, λ is a known scale factor, and E is the known external calibration that aligns world co-ordinates with the camera's frame of reference. As is standard in geometry, this formulation finds the single body pose that best explains all viewpoints; this is not equivalent to applying a single camera approach to each view and averaging the results.

Under this loss formulation, the camera terms are not independent but share a single 3D pose reconstruction Q_r, to which they contribute in equal measure.
Again, this can be directly solved as an unconstrained least squares problem given R; and as discussed in the previous subsection, we continue to marginalize over the space of rotations.

4.2.2.3 Robust losses for outlier rejection

Finally, the use of the squared Frobenius norm as in the previous section makes the reconstruction less robust to occlusions and to wrongly predicted joints. If the camera views were aligned, the first term of (4.5) would be minimized by a pose that averages over the different predictions. Use of the Frobenius norm would mean that if only one prediction is in the wrong place, it would "pull" the reconstruction

towards the mistake rather than discarding it as an outlier. Instead we replace the squared Frobenius norm with a Huber loss.

$$\underset{\tilde{Q}_R,s,a}{\arg\min}\,\lambda \sum_{c\in\mathscr{C}} ||I_c - \Pi E \tilde{Q}_R||_\varepsilon + ||\tilde{Q}_R - sR(\mu + a\cdot\mathbf{e})||_2^2 + \sigma^2\cdot a^2 \tag{4.6}$$

where the Huber Loss $||x||_\varepsilon = \sum_i |x_i|_\varepsilon$ and

$$|x|_\varepsilon = \begin{cases} \frac{|x_i|^2}{2} & \text{if } |x_i| \leq \varepsilon \\ \varepsilon|x_i| - \frac{\varepsilon^2}{2} & \text{otherwise.} \end{cases} \tag{4.7}$$

Although (4.6) is not a least square problem, it can be solved as an iterative re-weighted least squares problem (IRLS).

In practice, 5 iterations of least squares are sufficient to obtain a high quality solution. Although robust to outliers, this new loss remains convex given a choice of rotation, so local minima are not a concern.

As an implementation detail, the *IRLS* is solved by using a loop in which, at each step, the matrices involved in the least square problem are re-weighted by taking the rows corresponding to the re-projection loss term and multiplying them by weights determined by the contribution of each row to the residual during the previous step. As a consequence, this procedure has no impact on the gradient propagation during training and allows an end-to-end training as in the previous framework (see Sec. 3.2).

4.3 DATASETS

Data driven applications like the proposed approach require a large amount of data to be trained with. The larger the dataset the better the model can learn how to perform a specific task.
Similarly to what has been described in the previous chapter (see Sec. 3.3), we make use of the Human3.6M dataset [129] for training and evaluating the model.

Following the camera model and inference of the previous Chapter 3, we continue to assume a scaled orthographic model. Importantly, we assign the same choice of scale to all cameras. This assumption is noticeably stronger than the previous scaled orthographic reconstruction, which is equivalent to assuming that perspective distortions due to the varying depth of the object in any one frame can be safely ignored.

4.3.1 HUMAN3.6M

Our model was trained and tested on the *Human3.6M dataset*, which is one of the largest datasets consisting of 3.6 millions accurate 3D human poses [129], acquired by recording the performance of 5 female and 6 male actors, which wasgenerated in a multi-source capture studio with the ground-truth reconstructions coming from a ten camera Vicon studio, and four video cameras facing each another at right angles and far enough to fullycapture a 4 by 3 meter studio environment. (see Figure 4.4[2])

Unlike the previous chapter (see Sec. 3.3), during the training of our model, the four different camera views have been considered together in order to be able to exploit geometry, when the camera extrinsic parameters are known.

With the four cameras facing towards each other, our stronger assumption does not allow increase in overall scale due to movements towards one camera, as this would correspond with movement away from another camera and a corresponding

[2]Image taken from the Human3.6M dataset's website.

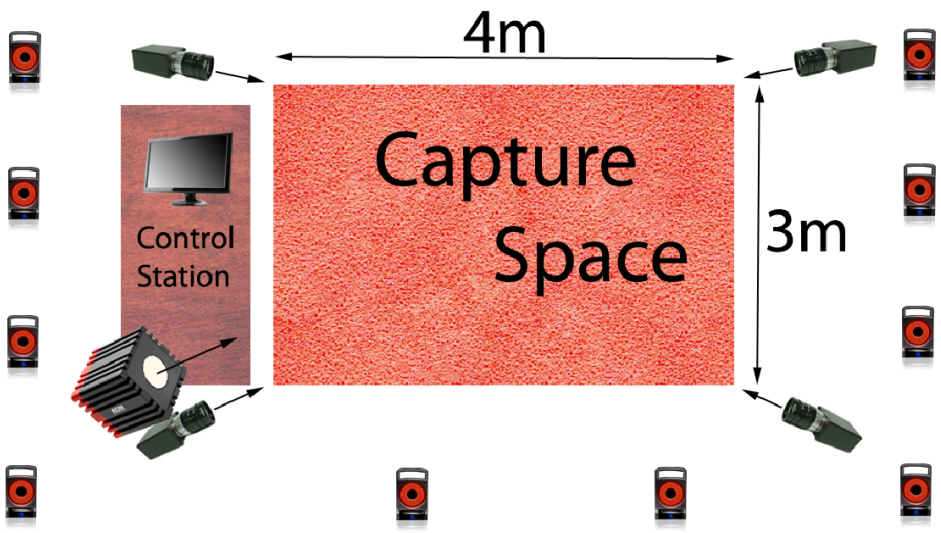


Figure 4.4: Human3.6M camera positions

decrease in scale. However, it does allow for changes in scale of the object itself allowing our algorithm to handle people of different sizes.

The proposed approach has been evaluated both in 2D and 3D on all the different actions, using different evaluation protocols (see Sec. 3.4).

4.3.2 CMU PANOPTIC DATASET

The CMU Panoptic Dataset by Joo *et al.* [145], consists of 65 sequences (5.5 hours), for a total 480 synchronized video streams of multiple people engaged in social activities, that have been used to produce the labeled time-varying 3D structure of anatomical landmarks on individuals in the space, for a total of 1.5 million 3D skeleton annotations.

The system used to capture the dataset (see Fig. 4.5 [3]) consists of:

- 480 VGA cameras, 640×480 resolution, 25 fps
- 31 HD cameras, 1920×1080 resolution, 30 fps

[3]Image taken from the CMU Panoptic Dataset website.

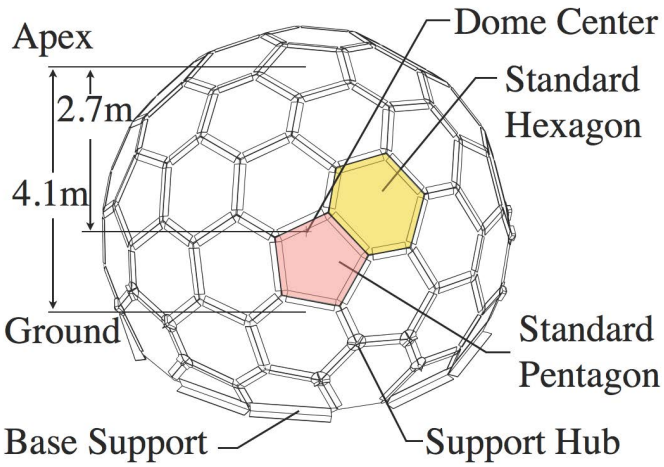


Figure 4.5: System developed for capturing the CMU Panoptic Dataset

- 10 Kinect 2 Sensors: 1920×1080 (RGB), 512×424 (depth), 30 fps
- 5 DLP Projectors, synchronized with HD cameras

Figure 4.6 shows the camera positions inside the dome.

An example of images contained in the dataset can be seen in Figure 4.7[4]. Among

[4]Images taken from the CMU Panoptic Dataset website.

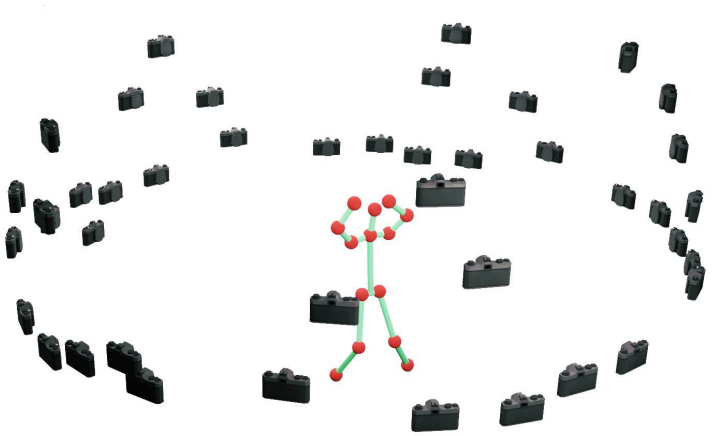


Figure 4.6: Camera placement in CMU Panoptic Dataset

Figure 4.7: Example images from the CMU Panoptic Dataset

those, we have only used a subset containing a single person per frame (Pose sequences).

4.4 DATA AUGMENTATION

One concern when trying to show how additional data can lead to improved results in the 3D reconstruction of people, is the restrictive form of the Human3.6M evaluation dataset. With the limited appearance and repetitive range of actions, that occur both in the training and in the evaluation sets, networks trained on more general datasets might perform worse than those trained on restrictive datasets that are closer to the test data. To avoid such issues, we make use of an additional set of actors performing the same actions captured by the authors of the Human3.6M dataset.

As with many datasets in computer vision, Human3.6M was originally subdivided into training, test and validation sub-sets; the reconstructions for the test-set were not made publicly available, to avoid over-fitting. However, for historic reasons, the test set has gone largely unused, with detailed evaluations being reported on the validation set. This means that we have access to a publicly available additional corpus, composed of unlabeled images from 2 men and woman[5], captured in the same environment.

To illustrate how 3D data gathered by our method can improve existing results, we augment two existing networks using this data. The produced results provided in section 4.5 show clear improvement over published results, and help make the case not just that better networks are needed for better results, but also more data.

Additional data can help 3D predictions in two separate ways, either *a)* by improving the 2D localization of joints, or *b)* by improving the 3D lifting from the same 2D inputs.
To show that our method returns results of sufficiently high quality to improve both components, we perform two separate experiments: *i)* we show improvements on 2D joint prediction while keeping the 3D lifting constant; *ii)* we show how a generic

[5]Human3.6M dataset does not provide video for subject S10.

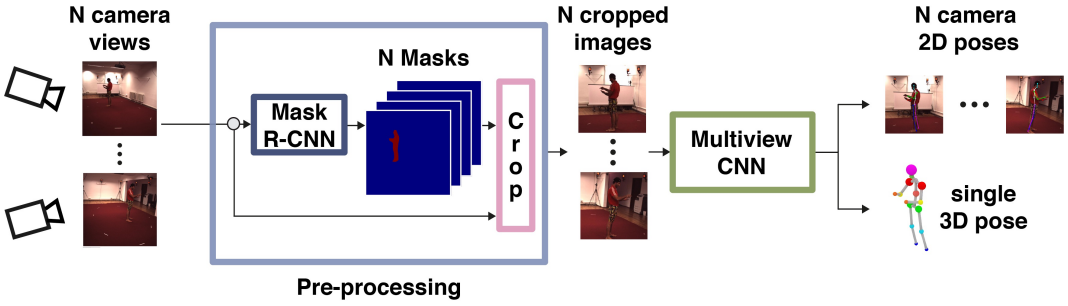


Figure 4.8: Labeling data using multi-camera 3D pose estimator

lifter that takes as input pre-computed joint locations can be improved by training on our additional 3D data.

4.4.1 LABELING DATA

The process of generating additional data by labeling the *official test-set* contained in the Human3.6M dataset is shown in Figure 4.8, where, given as input N sets of C camera RGB images

$$\{(I_1^1,\ldots,I_C^1),\ldots,(I_1^N,\ldots,I_C^N)\}$$

using the described multi-camera approach, the goal is to generate a set of 2D and 3D poses $\mathbf{Q}$

$$\mathbf{Q} = \{(\hat{p}_1^1,\ldots,\hat{p}_C^1,\hat{P}^1),\ldots,(\hat{p}_1^N,\ldots,\hat{I}_C^N,\hat{p}^N)\}$$

where $\hat{P}^j$ is the 3D pose expressed in world coordinates predicted using set of images $(I_1^j,\ldots,I_C^j)$, and $\hat{p}_i^j$ is the 2D pose resulting from the orthogonal projection of $\hat{P}^i$ over camera j.

Although conceptually simple, multiple small issues arise from most experiments reporting results on an automatically pre-processed version of the Human3.6M dataset.

First, images are independently run through the vanilla *Mask R-CNN* architecture

Figure 4.9: Mask R-CNN output

[146] (see Fig. 4.9), without any fine-tuning, in order to extract both the bounding box and the silhouette of the person represented in the images.

This information is essential for cropping the area of the image containing the person in a similar manner to what is done on images with ground truth 2D data, guaranteeing that: *i)* all the joints are inside the cropped region, centered around the center of mass of the person; *ii)* the aspect ratio is one and *iii)* 25 pixels of margin are added to the cropped region.

The approach used to identify the cropping area around the person is based on a heuristic, where the silhouette S identified by Mask R-CNN is firstly used to find the center R_c as

$$R_{c_x} = (\max(S_x) - \min(S_x))/2$$
$$R_{c_y} = (\max(S_y) - \min(S_y))/2$$

followed by the distance d, for which all the joints are contained inside the region, as well as the aspect ration is close to one

$$d_x = \max|S_x - R_{c_x}|$$
$$d_y = \max|S_y - R_{c_y}|$$
$$d = \max\{d_x, d_y\} + margin$$

where S_x, S_y are the vectors containing all the x and y coordinates of the silhouette expressed in pixels and *margin* corresponds to the 25 pixels margin added around the person.
Finally, the cropped region correspond to the rectangle whose top left (TL) corner and bottom right (BR) corner coordinates are

$$R^{TL} = (R_{c_x} - d, R_{c_y} - d)$$
$$R^{BR} = (R_{c_x} + d, R_{c_y} + d)$$

assuming $(0,0)$ is in the top-left corner of the image.

The result of applying this approach using as input the segmentation mask shown in Fig 4.9, where S corresponds to the red region, is shown in Figure 4.10.
These cropped regions are then used as inputs to our *multi-camera network* which estimates 2D body poses for each camera view and identifies the 3D pose most consistent with the set of 2D poses. Finally, the 3D pose is projected into 2D for

Figure 4.10: Cropping region from segmentation map

each camera view using the known camera calibration.

Data labeled by our approach is used to extend existing datasets. We simply treat the predicted bounding-boxes, 2D landmarks and 3D reconstructions the same way as existing ground truth training data.

4.4.2 SEMI-SUPERVISED LEARNING

Using the novel multi-camera pose estimator described in the previous Section 4.2.2, we can now augment the corpus used for training the 2D pose detector where, when no ground-truth data exists, we can train using a 3D pose that minimizes the 2D re-projection error with respect to the 2D multi-view predictions.

This gives us a unified framework for training a CNN, where we solve for 2D predictors $\mathbf{x}_c$ that take a set of images $\mathbf{I}_f$ as input to minimize the following training

loss:

$$\arg\min_{\mathbf{x}} \sum_{f\in\mathscr{S}} \sum_{c\in\mathscr{C}} \ell\left(\mathbf{x}_c(\mathbf{I}_f), g_{(f,c)}\right) + \sum_{f\in\mathscr{U}} \sum_{c\in\mathscr{C}} \ell\left(\mathbf{x}_c(\mathbf{I}_f), P_c\left(R(\mathbf{x}(\mathbf{I}_f))\right)\right) \tag{4.8}$$

where $\mathbf{I}_f$ is a set of images associated with the full set of cameras for a particular frame f and $\mathbf{x}(\mathbf{I}_f)$ is the collection of 2D estimates of joint locations for each camera $c \in \mathscr{C}$, where $\mathscr{C}$ is the set of cameras of known calibration. This loss function can be broken down into two main components:

- For supervised frames, $\mathscr{S}$, that have 2D ground-truth of joint locations $g_{(f,c)}$ associated with them, we apply as loss a mse-loss function between predictions and ground truth positions
- For unsupervised frames $\mathscr{U}$, we induce the same loss between the projection into a particular cameras viewpoint $P_c(\cdot)$ of a *unified reconstruction* $R(\mathbf{x}(\mathbf{I}_f))$ generated from the set of estimated 2D poses of across all cameras.

The definition of $\mathbf{x}$ is inherently somewhat involved, as it refers to a multistage estimator that makes per-camera 2D estimates of joints which are fused together into a coherent 3D pose at each stage, which is then refined in the 2D pose estimation of the next stage. This refinement is performed per camera, and means that the joint estimation of each joint in any camera is distinct from the projection of the 3D estimated point.

The question is what form does R — a function of the 2D poses — need to have to guarantee convergence? We first note that choosing R as an arbitrary regression function that maps from 2D points to 3D (e.g. [82]) does not imply convergence. The reason for this is that arbitrary regression functions are powerful enough to correct for systematic biases in the 2D estimation, such as incorrectly offset points or a shrinking bias that causes points to be estimated closer to the centroid than it should be. Using such an R can lead to divergent estimates as on unsupervised data; the neural network updates so that $\mathbf{x}_c$ equals $P_c(R(\mathbf{x}(\mathbf{I}_c)))$, leading to a new estimate of R that drifts from its previous position.

In contrast, if we choose R to be an implicit minimizer defined over a combination of a re-projection loss and a regularization term i.e.

$$R(\mathbf{x}_c, z) = \arg\min \ell(\mathbf{x}_c, P_c(z)) + \text{reg}(z) \tag{4.9}$$

with x_c the image from camera c, $P_c(.)$ the projection of the 3D pose z onto camera c, and *reg* a regularizer that enforce bone length preservation. Optimizing on $\mathbf{x}$ is guaranteed to converge as iteratively updating x using gradient descent and then re-estimating for z equates to hill-climbing optimization where we jointly optimize:

$$\min_{\mathbf{x}} \left(\sum_{f \in \mathscr{S}} \sum_{c \in \mathscr{C}} \ell\left(\mathbf{x}_c(\mathbf{I}_f), g_{f,c}\right) + \sum_{f \in \mathscr{U}} \min_{z} \sum_{c \in \mathscr{C}} \ell\left(\mathbf{x}_c(\mathbf{I}_f), P_c\left(R(\mathbf{x}(\mathbf{I}_f), z)\right)\right) \right) \tag{4.10}$$

The challenge is that the estimates $\mathbf{x}_c$ are sparse co-ordinates induced by taking the arg max over a heatmap. Here we make use of a variant of the sub-gradient approach of [16], and note that if $\mathbf{x}_c$ is not in a valid location a sub-gradient of the arg max can be obtained by decreasing the heatmap around one of its maximal values and increasing it at any non-maximal location.

We choose to increase the heatmap at the point $P_c(R(\mathbf{x}(\mathbf{I}_f)))$ as it drives convergence faster towards a zero re-projection error. This is equivalent to the standard updates of [15], where the update is the difference between current heatmap and a ground-truth heatmap induced by the true 2D locations, but instead of using the true 2D locations, we use the projection of the estimated 3D pose. As with [15, 16] to avoid vanishing gradients this loss is imposed at all stages.

4.5 EXPERIMENTAL EVALUATION

This section describes the various evaluation protocol used on the different performance comparisons with other state-of-the-art approaches. Following, the reconstructed poses are analyzed to asses the quality of the reconstructions.
Finally, an analysis over the loss function effects is provided.

4.5.1 EVALUATION PROTOCOLS

2D Evaluation: since the goal of the approach is 3D Human Pose Estimation, we limit our evaluation of 2D performance to measuring the pixel distance of predicted joints with their corresponding ground truth location. Therefore, what we refer to as 2D pixel error is defined as:

$$error = \frac{1}{N}\frac{1}{L}\sum_{n}^{N}\sum_{l}^{L}||\hat{P}_l^n - P_l^n||_2$$

where N is the number of frames, L is the number of joints, P_l^n is the ground truth position of joint l for the n-th frame, and $\hat{P}_l^n$ is its predicted location.

3D Evaluation: Several evaluation protocols have been followed by different authors to measure the performance of their 3D pose estimation methods on the Human3.6M dataset.

Protocol 1, the most standard evaluation protocol on Human3.6M was followed by [129, 66, 136, 68, 137, 86, 81]. The training set consists of 5 subjects (S1, S5, S6, S7, S8), while the test set includes 2 subjects (S9, S11). The original frame rate of 50 fps is down-sampled to 10 fps and the evaluation is on sequences coming from all 4 cameras and all trials. The error metric is the Euclidean distance from the estimated 3D joints to the ground truth, averaged over all 17 joints of the Human3.6M skeletal model, without performing any data alignment.

Protocol 2, followed by [6], it selects the same subjects for training and testing as *Protocol 1*. However, evaluation is only on sequences from trial 1 and the original video is not sub-sampled. The error metric used in this case is the the average per-joint 3D error after aligning the reconstruction with the ground-truth using Procrustes analysis, averaged over 14 joints.

4.5.2 QUANTITATIVE RESULTS

The comparison of the proposed multi-camera approach with other state-of-the-art techniques (both monocular and multi-view) under *protocol 1* is shown in Table 4.1. Our proposed approach outperforms monocular methods, reducing the error by over 10 millimeters, and gives better results than the best multi-camera method of Pavlakos *et al.* [110] with an improvement of more than 4 millimeters.

We also create a novel baseline based on generating monocular reconstructions from each view using the method of Martinez *et al.* [82] and averaging them after alignment. [82] was chosen due to its great performance coming from the data preprocessing step, which uses predicted 2D joint locations as input to estimate the 3D pose. This performs almost as well as Pavlakos *et al.*, and is reported in table 4.1 as "Multi-view Martinez".

Similarly, Table 4.2 shows a comparison with other state of the art approaches using *protocol 2*, when the entire set of four camera views is used. The approach is assessed for both skeleton definitions consisting of a total of 14 and 17 joints.

Having a multi-camera system as the one provided by the Human3.6M dataset (see Sec 4.3), it is interesting to run the same set of experiments run earlier on a sub-set of cameras, to identify how their position is affecting the overall results, shown in Table 4.3.

It is particularly noticeable how some multi-camera placements are more effective than others, especially when using a small number of cameras (two in this

experiment). When evaluating camera whose positions are opposite to each other ($\{cam1, cam4\}$ and $\{cam2, cam3\}$), the performance drops.

4.5.2.1 Labeling data

Given the noticeable improvement in accuracy obtained by using multiple cameras rather than just one, we have then assessed the ability of the proposed multi-view 3D human pose estimator to meaningfully label unlabeled data that can be used at train time in order to achieve better performance.

First test consists in evaluating both 2D and 3D reconstruction errors when unlabeled data are used during training of monocular 3D pose estimators. Specifically, the approach proposed in Chapter 3 and the method introduced by Martinez *et al.* [82] have been evaluated on a variety of experiments where the models were trained using ground-truth training data provided by the Human3.6M dataset [129], and additional unlabeled data (Subjects $\{S2, S3, S4\}$), automatically labeled as previously described in section 4.4. Results of the training are shown in Table 4.4. Note that in both approaches, the original training hyper-parameters were used and the respective models have been retrained using the augmented training data, without performing any hyper-parameters tuning (results can be improved by performing

Monocular	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
LinKDE [129]	132.7	183.6	132.4	164.4	162.1	205.9	150.6	171.3	151.6	243.1	162.1	170.7	177.1	96.6	127.9	162.1
Li *et al.* [66]	-	136.9	96.9	124.7	-	168.7	-	-	-	-	-	-	132.1	69.9	-	-
Tekin *et al.* [136]	102.4	158.5	87.9	126.8	118.4	185.1	114.7	107.6	136.2	205.7	118.2	146.7	128.1	65.9	77.2	125.3
Zhou *et al.* [80]	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Tome *et al.* [16]	64.9	73.5	76.8	86.4	86.3	110.7	68.9	74.8	110.2	172.9	84.9	85.8	86.3	71.4	73.1	88.4
Pavlakos *et al.* [70]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Tekin *et al.* [69]	53.9	62.2	61.5	66.2	80.1	79.5	64.6	83.2	70.9	107.9	70.4	68.0	77.8	52.8	63.1	70.8
Katircioglu *et al.* [147]	54.9	63.3	57.3	62.3	70.3	77.4	56.7	57.1	79.0	97.1	64.3	61.9	67.1	49.8	62.3	65.4
Zhou *et al.* [73]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.15	66.05	51.4	63.2	55.3	64.9
Martinez *et al.* [82]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Multi-view	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Multi-View Martinez	46.5	**48.6**	54.0	51.5	67.5	70.7	48.5	49.1	69.8	**79.4**	57.8	53.1	56.7	42.2	45.4	57.0
PVH-TSP [109]	92.7	85.9	72.3 7	93.2	86.2	101.2	75.1	78.0	83.5	94.8	85.8	82.0	114.6	94.9	79.7	87.3
Pavlakos *et al.* [110]	**41.2**	49.2	42.8	**43.4**	55.6	**46.9**	**40.3**	63.7	97.6	119.0	52.1	**42.7**	51.9	**41.8**	**39.4**	56.9
Ours	43.3	49.6	**42.0**	48.8	**51.1**	64.3	**40.3**	**43.3**	**66.0**	95.2	**50.2**	52.2	**51.1**	43.9	45.3	**52.8**

Table 4.1: Evaluation of multi-view 3D pose estimator on Human3.6M dataset using Protocol 1 compared to other approaches.

Protocol 2	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Akhter & Black [51] 14j	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7	173.7	177.8	181.9	176.2	198.6	192.7	181.1
Ramakrishna *et al.* [50] 14j	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6	175.6	160.4	161.7	150.0	174.8	150.2	157.3
Zhou *et al.* [79] 14j	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1	137.5	106.0	102.2	106.5	110.4	115.2	106.7
Bogo *et al.* [6] 14j	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Tome *et al.* [16] 14j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	79.6
Moreno-Noguer [83] 14j	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Ours 14j	**40.4**	**42.8**	**39.8**	**44.8**	**47.5**	**59.1**	**36.6**	**37.0**	**55.8**	**82.3**	**46.8**	**48.9**	**48.2**	**38.8**	**40.4**	**47.6**
Pavlakos *et al.* [70] 17j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	51.9
Martinez *et al.* [82] 17j	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	**69.4**	49.2	**45.0**	49.5	38.0	43.1	47.7
Ours 17j	**38.2**	**40.2**	**38.8**	**41.7**	**44.5**	**54.9**	**34.8**	**35.0**	**52.9**	75.7	**43.3**	46.3	**44.7**	**35.7**	**37.5**	**44.6**

Table 4.2: Evaluation of multi-view 3D pose estimator on Human3.6M dataset using Protocol 2 compared to other approaches. Comparison is shown for both skeleton definitions (14 and 17 joints version)

Protocol 1	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Cam 1 2	56.9	60.6	53.6	57.3	62.7	78.5	49.9	52.2	74.2	114.5	60.2	65.9	59.7	55.0	57.6	64.24
Cam 1 3	55.6	58.2	54.8	60.9	66.4	81.5	54.4	53.6	80.6	125.8	63.0	65.2	62.5	55.9	60.1	66.70
Cam 1 4	68.9	71.4	63.5	82.8	82.5	109.5	70.4	78.7	102.4	130.2	76.1	77.8	73.1	64.2	68.8	81.18
Cam 2 3	69.8	72.9	63.9	83.1	72.9	101.4	66.9	60.8	112.4	123.7	74.2	79.1	71.9	63.6	69.3	79.37
Cam 2 4	50.8	63.8	49.3	56.6	57.9	78.1	47.0	49.7	65.5	113.0	56.0	66.1	63.5	58.1	57.4	62.41
Cam 3 4	49.9	59.2	50.7	59.2	61.2	76.8	47.4	52.0	81.2	111.3	59.9	63.4	62.0	53.7	56.1	63.33
Protocol 2 14j	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Cam 1 2	48.8	49.6	50.6	53.0	55.3	70.6	45.8	43.6	66.2	93.6	56.3	58.0	54.0	49.6	50.7	56.59
Cam 1 3	48.2	51.0	52.1	54.7	60.5	75.1	46.9	49.4	70.9	100.0	58.6	59.3	57.7	49.4	51.8	59.23
Cam 1 4	57.0	57.4	56.9	69.4	67.3	97.3	53.1	58.5	75.7	106.2	61.4	63.9	65.7	56.2	62.8	66.71
Cam 2 3	60.2	55.9	55.6	68.7	62.8	81.0	51.3	49.7	76.1	99.2	61.5	64.2	63.7	57.8	61.4	64.52
Cam 2 4	47.0	50.2	42.8	51.2	54.6	68.7	41.3	42.8	59.5	99.1	52.1	54.3	57.0	49.2	48.8	54.78
Cam 3 4	47.5	52.1	47.7	54.0	58.0	68.3	42.3	42.9	65.0	97.2	54.6	55.6	58.3	48.5	50.8	56.67
Protocol 2 17 j	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Cam 1 2	45.5	46.4	48.5	49.2	51.3	65.5	42.9	40.7	61.7	85.4	51.6	54.4	50.0	45.2	46.3	52.51
Cam 1 3	45.0	47.6	49.6	50.7	55.8	69.5	44.2	46.4	66.1	91.5	53.9	55.9	53.2	45.2	47.4	54.94
Cam 1 4	53.1	53.6	54.0	64.0	62.5	90.4	50.0	55.4	71.0	97.6	56.9	59.7	60.1	51.1	57.1	61.93
Cam 2 3	56.5	52.2	53.7	64.0	58.6	75.7	48.0	46.7	71.8	91.0	56.2	60.3	59.0	52.7	55.9	60.07
Cam 2 4	44.3	47.2	41.2	47.7	50.7	63.7	39.5	40.8	55.8	90.6	48.0	50.9	52.7	45.2	44.8	51.05
Cam 3 4	44.6	48.7	45.6	50.2	53.5	63.2	39.8	40.3	61.3	89.0	50.2	52.3	53.8	44.1	46.5	52.64

Table 4.3: Two camera evaluation on Human3.6M dataset

grid-search).

The authors of [82] no longer have access to the retrained stacked-hourglass 2D networks that they take as an input, so we can not compute their 2D joint estimations on the held-out unlabeled data. Instead we repeat their experiments, by training the network using the 2D poses estimated by our monocular approach (see Chapter 3) as input, and using these inputs to drive the 3D prediction. Without optimizing

the hyper-parameters, this leads to a noticeable decrease in the performance of the algorithm over that reported by their paper, even though our 2D pose estimator has a lower 2D error than that of Martinez *et al.* reported in Tab 4.1 4.2.
Despite this, we still observe a substantial improvement in the 3D reconstruction from using more data. Note that for this experiment, we do not update the 2D pose estimations, and all improvement comes from the updated 3D estimator.

To illustrate that our method also improves 2D joint localization, we also retrain our own monocular approach, where as an initial step in training the algorithm, we computed a shape basis from MoCap data. This basis is not updated during the end-to-end training of the pose estimator, and the network itself is trained to improve 2D loss in joint predictions, returning a 3D pose as a side-effect of its 2D pose computation. Although we could update the 3D basis using our newly labeled data, we restrict ourselves to only updating the 2D pose predictor. As can be seen in table 4.4, this leads to a significant improvement in 2D error, and a corresponding reduction in the 3D error.

The noticeable improvement due to the newly labeled data, needs to be verified further, by proving that this is also happening when using multi-view 3D pose estimators. For this reason, the new multi-view architecture has been trained on a small subset of the Panoptic Dataset (see Sec 4.3) as used as an initialization when further training the model with *a)* Train-set (S) as ground truth data from the Human3.6M

Approach	Experiment	Human3.6M dataset		Δ	%
		Train	Train + new data		
Tome *et al.* [16]	3D error (P#1)	88.4 mm	84.4 mm	4.0	4.52
	3D error (P#2)	70.7 mm	67.2 mm	3.5	4.95
	2D error	9.5 pix	8.6 pix	0.9	9.47
Martinez *et al.* [82]	3D error (P#1)	75.8 mm	72.5 mm	3.3	4.35
	3D error (P#2)	57.6 mm	55.9 mm	1.7	2.95

Table 4.4: Evaluation on Human3.6M dataset of monocular approaches with unlabeled data used for training the models.

Protocol 1	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
MONOCULAR with 3D supervision																
LinKDE [129]	132.7	183.6	132.4	164.4	162.1	205.9	150.6	171.3	151.6	243.1	162.1	170.7	177.1	96.6	127.9	162.1
Li *et al.* [66]	-	136.9	96.9	124.7	-	168.7	-	-	-	-	-	-	132.1	69.9	-	
Tekin *et al.* [136]	102.4	158.5	87.9	126.8	118.4	185.1	114.7	107.6	136.2	205.7	118.2	146.7	128.1	65.9	77.2	125.3
Zhou *et al.* [80]	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Pavlakos *et al.* [70]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Tekin *et al.* [69]	53.9	62.2	61.5	66.2	80.1	79.5	64.6	83.2	70.9	107.9	70.4	68.0	77.8	52.8	63.1	70.8
Katircioglu *et al.* [147]	54.9	63.3	57.3	62.3	70.3	77.4	56.7	57.1	79.0	97.1	64.3	61.9	67.1	49.8	62.3	65.4
Zhou *et al.* [73]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.15	66.05	51.4	63.2	55.3	64.9
Martinez *et al.* [82]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
MULTI-VIEW																
PVH-TSP [109]	92.7	85.9	72.3 7	93.2	86.2	101.2	75.1	78.0	83.5	94.8	85.8	82.0	114.6	94.9	79.7	87.3
Pavlakos *et al.* [110]	**41.2**	49.2	42.8	**43.4**	55.6	**46.9**	40.3	63.7	97.6	119.0	52.1	**42.7**	51.9	41.8	**39.4**	56.9
Ours																
Train-set (**S**)	43.3	49.6	42.0	48.8	51.1	64.3	40.3	43.3	66.0	95.2	50.2	52.2	51.1	43.9	45.3	52.8
Train-set (**U**)	53.5	60.8	53.8	60.5	69.0	71.9	46.0	57.0	87.2	129.6	66.2	64.4	67.0	58.3	62.0	67.7
Unlabeled-set (**U**)	56.1	63.1	56.7	63.3	71.7	75.1	49.1	59.3	94.0	140.1	69.5	67.1	69.7	60.6	67.9	71.4
Train-set (**S**) + Unlabeled-set (**U**)	43.9	50.0	**41.7**	**46.8**	**49.1**	**61.3**	**36.2**	**39.2**	**64.5**	100.4	**49.6**	50.2	**47.9**	**41.0**	**43.1**	51.6
Train (**S**) + Unlabeled (**U**) + Evaluation (**U**)	**42.7**	**46.6**	41.8	47.3	51.2	61.9	38.3	44.6	65.0	**90.0**	50.6	**49.5**	49.9	43.1	44.3	**51.4**

Table 4.5: Evaluation of the proposed multi-view 3D pose estimator against its competitors on the Human3.6M dataset. The reported 3D pose error results are expressed in mm using the metric defined in Protocol 1. All methods in the top part of the table are monocular approaches supervised with ground truth 3D poses, while [109, 110] and **Ours** are multi-camera approaches. **U** and **S** stand for *Unsupervised* and *Supervised* respectively; where not indicated, assume supervised training.

train-set, *b)* Train-set (U) as data from the Human3.6M train-set labeled using our labeling process, *c)* Unlabeled-set (U) as data from the Human3.6M original test-set labeled using our labeling process, and finally *d)* as a combination of labeled and unlabeled sets.

The results of these trained models are shown in Table 4.5 and Table 4.6 where they are assessed using protocol 1 and protocol 2 respectively.

We assess again the effect of the labeled data on the monocular approach described in Chapter 3 considering a different amount of labeled data. Results are shown in Table 4.7.

4.5.2.2 Loss function

Finally, we explore the importance of the changes to the pose estimator made in Sec 4.2.1; particularly the use of a more robust Huber loss in place of the squared

Protocol 2	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Avg**
Akhter & Black [51] 14j	199.2	177.6	161.8	197.8	176.2	186.5	195.4	167.3	160.7	173.7	177.8	181.9	176.2	198.6	192.7	181.1
Ramakrishna *et al.* [50] 14j	137.4	149.3	141.6	154.3	157.7	158.9	141.8	158.1	168.6	175.6	160.4	161.7	150.0	174.8	150.2	157.3
Zhou *et al.* [79] 14j	99.7	95.8	87.9	116.8	108.3	107.3	93.5	95.3	109.1	137.5	106.0	102.2	106.5	110.4	115.2	106.7
Bogo *et al.* [6] 14j	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Tome *et al.* [16] 14j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	79.6
Moreno-Noguer [83] 14j	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Ours 14j																
Train-set (**S**)	40.4	42.8	**39.8**	44.8	47.5	59.1	36.6	**37.0**	**55.8**	82.3	46.8	48.9	48.2	38.8	40.4	47.6
Train-set (**U**)	50.2	56.1	52.3	56.1	66.1	69.9	42.8	46.9	70.9	112.1	61.4	62.7	63.4	55.0	58.6	62.4
Unlabeled-set (**U**)	52.5	58.7	55.6	59.3	68.6	72.0	46.9	48.9	75.2	118.5	65.0	64.3	66.7	58.0	64.2	65.7
Train (**S**) + Unlabeled (**U**)	41.0	45.0	40.4	43.5	**46.8**	58.5	34.5	**37.0**	62.4	94.8	48.6	**46.5**	**46.3**	**36.9**	**37.6**	48.7
Train (**S**) + Unlabeled (**U**) + Evaluation (**U**)	**39.1**	**41.8**	40.5	**42.9**	47.8	**58.1**	**33.8**	37.1	56.2	**80.5**	**46.4**	47.7	47.9	38.3	39.9	**46.9**
Pavlakos *et al.* [70] 17j	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	51.9
Martinez *et al.* [82] 17j	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	**69.4**	49.2	45.0	49.5	38.0	43.1	47.7
Ours 17j																
Train-set (**S**)	38.2	40.2	38.8	41.7	44.5	54.9	34.8	**35.0**	**52.9**	75.7	43.3	46.3	44.7	35.7	37.5	44.6
Train-set (**U**)	46.5	51.5	49.3	51.5	61.0	64.9	39.7	43.9	67.0	101.6	56.1	59.3	57.7	49.4	53.2	57.5
Unlabeled-set (**U**)	48.2	53.7	52.0	54.2	63.1	67.0	43.1	46.0	71.1	107.1	59.2	60.4	60.5	51.9	58.0	60.4
Train (**S**) + Unlabeled(**U**)	38.3	41.7	**38.4**	40.3	**43.2**	54.5	**32.2**	35.6	57.7	86.5	44.6	**44.5**	**42.7**	**33.7**	**34.6**	45.1
Train (**S**) + Unlabeled (**U**) + Evaluation (**U**)	**37.1**	**39.5**	39.6	**40.1**	45.0	**54.2**	**32.2**	35.1	53.1	**74.3**	**43.2**	45.8	44.6	35.3	37.0	**44.1**

Table 4.6: Evaluation of the proposed multi-view 3D pose estimator against other approaches using evaluation *Protocol 2* on the Human3.6M dataset. Note that all other methods are monocular. The $14j/17j$ annotation indicates the number of joints used in evaluation.

Training	2D Error	3D Protocol 2 14j
Panoptic(S)	25px	104.8 mm
Panoptic init. + Human3.6M train(S)	8.8 px	48.7 mm
Panoptic init. + Human3.6M train (S) + unlabeled. (U) + eval(U)	8.17 px	44.1 mm

Table 4.7: Evaluation of the monocular approach (see Chapter 3) using different amount of labelled data. Note: since Panoptic 2D labels are a sub-set of the Human3.6M's 2D poses, using only Protocol 2 on 14 joints. **U** and **S** stand for *Unsupervised* and *Supervised* respectively.

Frobenius norm, (Eq. 4.5 and Eq. 4.6). The reconstruction error for different variants of our approach is shown in Table 4.8.

Huber loss (2 cameras) shows the mean and standard deviation of the reconstruction using only a pair of cameras at right angles with one another. GT Orthographic Triangulation shows the error due to the use of an orthographic camera, i.e. the the reconstruction error given perfect detections.

Although, many works make use of the Huber loss as a more stable approximation of the ℓ_1 norm, this is not the case for us. Upon inspection, we found that the optimal choice of ε that resulted in the lowest 3D reconstruction error treated half

Formulation	Error *Protocol 1*	Error *Protocol 2*
Squared Frobenius (no averaging)	59.6 mm	51.1 mm
Squared Frobenius	59.4 mm	51.8 mm
Huber loss	52.8 mm	44.6 mm
Huber loss (2 cameras)	64.2	52.8
GT Orthographic Triangulation	27.9 mm	20.7 mm

Table 4.8: Pose estimator variations

of the joints with ℓ_1 norm and the other half with the squared Frobenius norm which confirms that the Huber loss is effectively used to weigh the relevance of each joint on a case by case basis.

A small improvement can also be seen from marginalizing over the rotations, although this modification primarily improves the stability of reconstructions rather than reducing the overall error. Finally we show how much error can be attributed to the camera model, by triangulating ground-truth detections under orthographic assumptions. This is reported as "GT Orthographic Triangulation".

4.5.3 QUALITATIVE RESULTS

Figure 4.11 shows some sampled 2D and 3D poses with the respective reconstruction error for some multi-camera frames taken from the test-set of Human3.6M dataset. The sorted error plot is based on sampling the error every 10^{th} frame of trial 1. Ground-truth reconstructions are given in blue, and the rows labelled protocol 1 and protocol 2 both show the same reconstructions in red, however protocol 1 shows the reconstruction *unaligned* with the ground-truth, and protocol 2 shows the reconstruction *aligned* to the ground-truth.

A similar experiment is in Figure 4.12 where the difference between fully-supervised and semi-supervised performance is shown.

An additional qualitative result that is expected but equally important to demonstrate

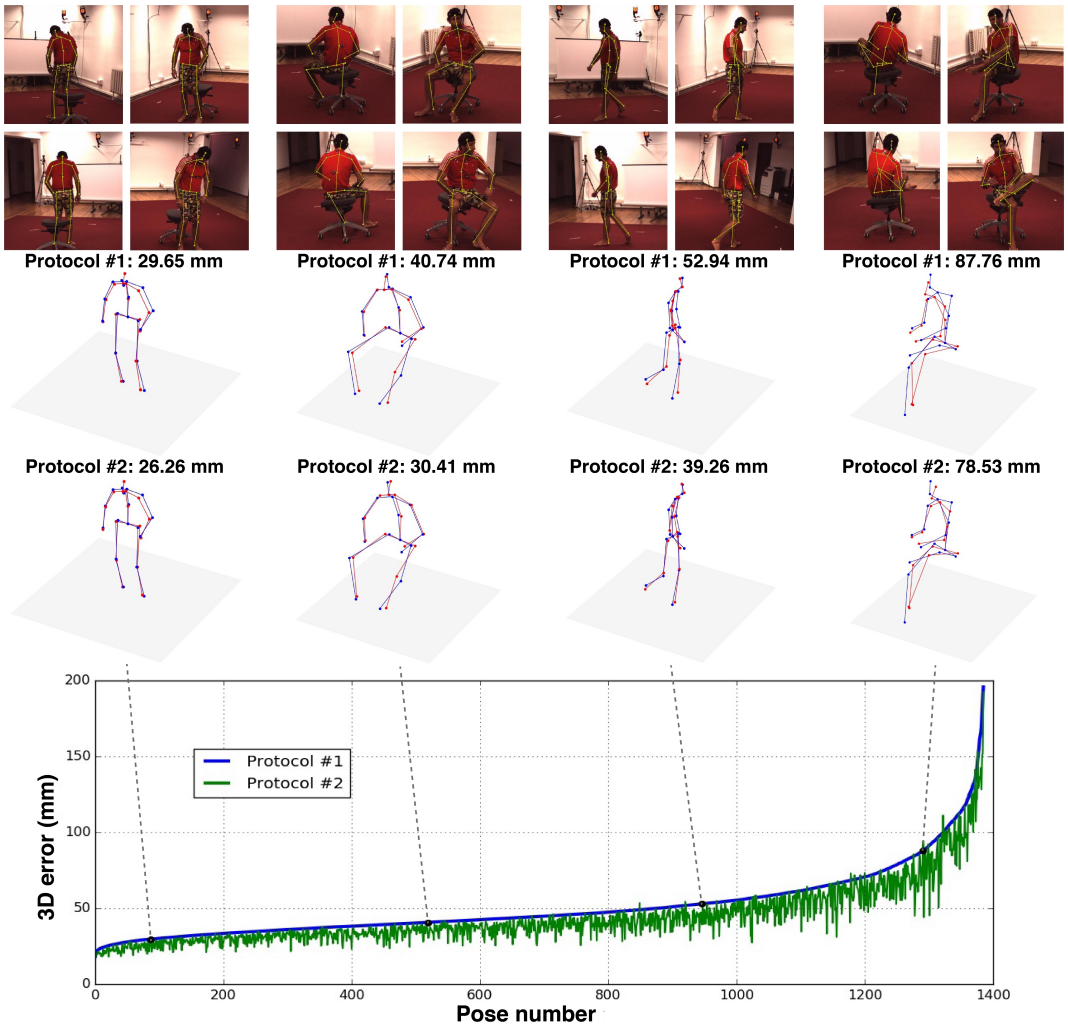


Figure 4.11: Multi-view 3D pose reconstructions on Human3.6M dataset

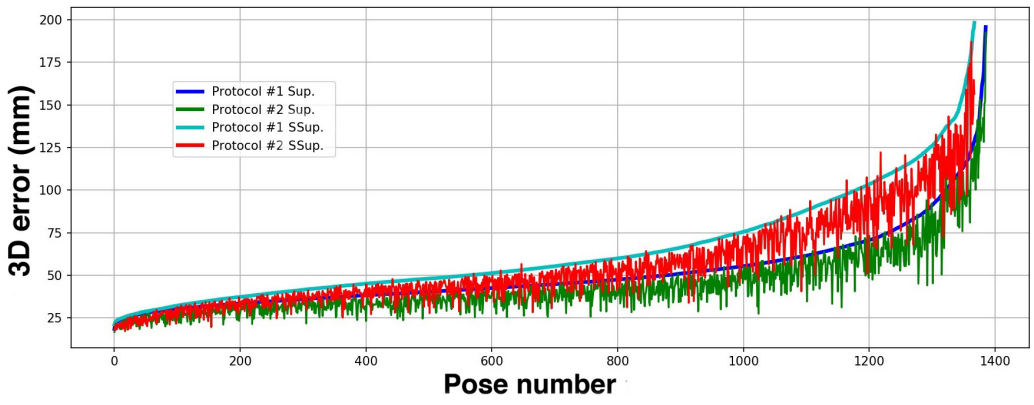
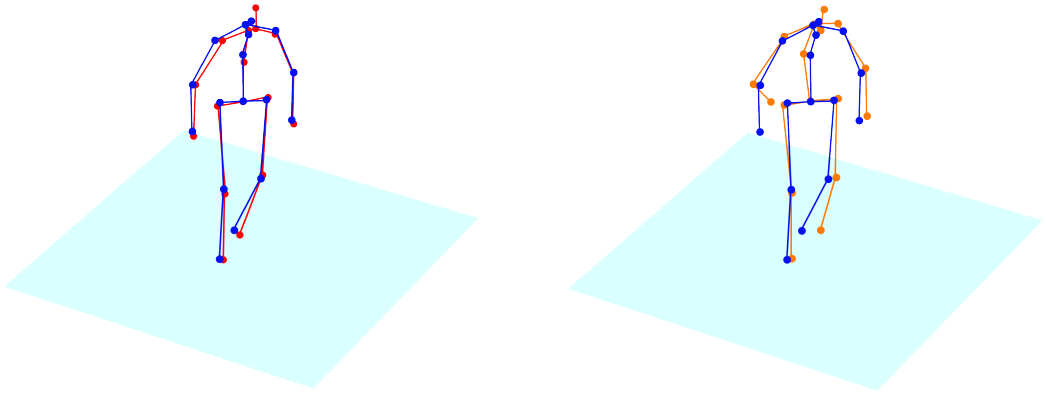


Figure 4.12: Supervised vs. Semi-Supervised multi-view reconstructions

is the visual difference between predictions computed using the multi-view model versus predictions using a monocular approach. Please refer to Figure 4.13.



a) Multi-view 3D pose prediction
(red - predicted pose, blue - ground truth pose)

a) Monocular 3D pose prediction
(orange - predicted pose, blue - ground truth pose)

Figure 4.13: Single vs. Monocular reconstruction against ground truth pose. Notice how multi-view approach produces a more accurate pose.

Finally, in Figure 4.14 we showcase how the multi-view approach is able to deal with multiple mispredictions on one or multiple joints simultaneously. Particularly, notice how in figure 4.14*a)* and 4.14*b)* can deal with one or even two cameras failing in predicting the correct joint location, and still be able to predict the correct pose (pose with a small error compared to the ground truth one). In 4.14*c)* three out of four cameras wrongly estimate the joint and therefore the left arm is not correctly estimated in 3D, as one would expect.

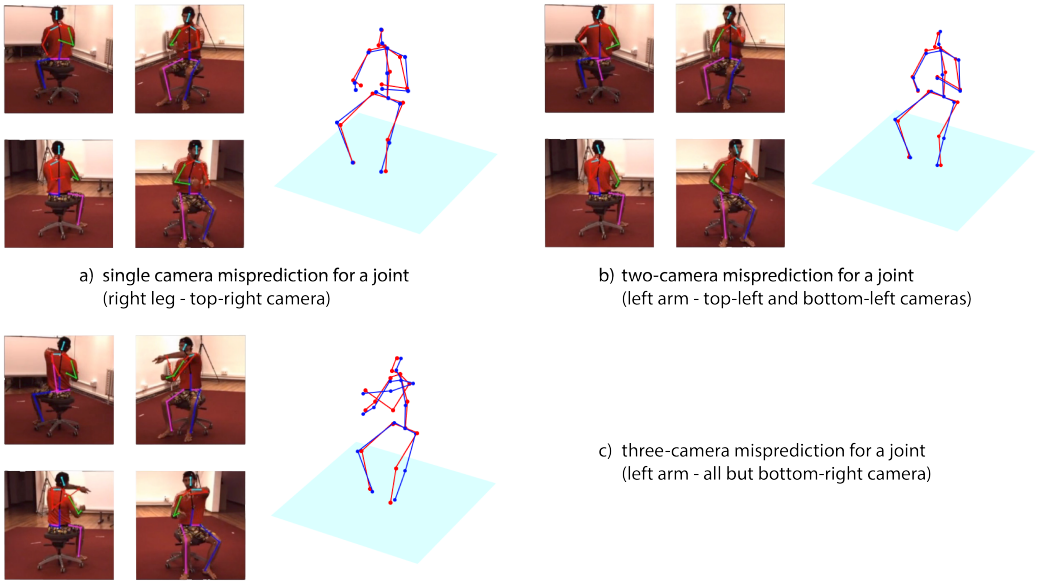


Figure 4.14: Robustness to mispredictions.
Notice how multi-view approach produces a more accurate pose.

4.6 CONCLUSION

In this chapter, inspired by the work by [16, 15, 110] we have presented a novel "hybrid pipeline approach" for marker-less multi-camera motion-capture with a multi-stage architecture that allows us to recover from initial misdetections, and still make use of image cues in locating joints in subsequent stages.

We have demonstrated the clear benefits and robustness of our approach by noticeably improving over existing multi-view marker-less motion capture system, achieving state-of-the-art results that allow this approach to be used in more accurate 3D human pose detection applications than what it could be achieved using monocular approaches.

Finally, we have demonstrated how such method can be used to improve the performance of "standard" monocular approaches by labeling additional data that can be used at running time, with the advantage of decreasing the reconstruction error while at the same time using a single monocular approach at inference time.

CHAPTER 5

EGOCENTRIC HUMAN POSE ESTIMATION

5.1 OVERVIEW

The advent of VR and AR technologies have led to a wide variety of applications in areas such as entertainment, communication, medicine, CAD design, art, and workspace productivity. These technologies mainly focus on immersing the user into a virtual space by the use of a head mounted display (HMD) which renders the environment from the very specific point of view of the user. However, current solutions have been focusing so far on the video and audio aspects of the user's perceptual system, leaving a gap in the touch and proprioception senses. Partial solutions to the proprioception problem have been limited to hands whose positions are tracked and rendered in real time by the use of controller devices. The 3D pose of the rest of the body can be inferred from inverse kinematics of the head and hand poses [148], but this leads to inaccurate estimates of the body configuration with a large loss of signal which impedes compelling social interaction [149] and even lead to motion sickness [150].

In this paper we present a novel approach for full-body 3D human pose estimation *from a monocular camera installed on a HMD* (see Fig. 5.1). In our novel solution, the camera is mounted on the rim of a HMD looking down, effectively just 2.1 centimeters away from an average size nose. With this unique camera viewpoint, most of the lower body appears self-occluded (see Fig. 5.2). In addition, the strong per-

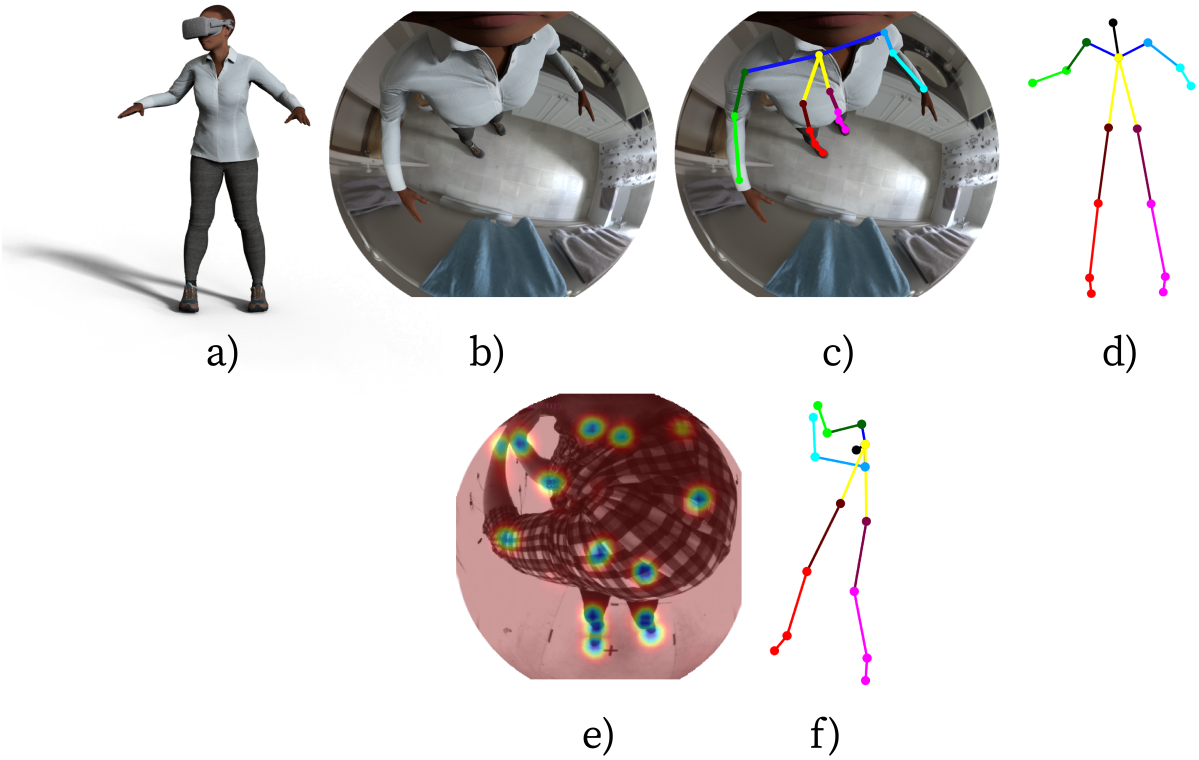


Figure 5.1: Egocentric Human Pose Estimation: *(a)* external camera viewpoint showing a synthetic character wearing the headset; *(b)* example image rendered from the egocentric camera perspective with the person placed in a photo-realistic environment; *(c)* 2D and *(d)* 3D poses estimated with our algorithm; *(e)* real image acquired with our HMD-mounted camera with predicted 2D heatmaps; and *(f)* the estimated 3D pose, showing good generalization to real images.

spective distortion, due to the fish-eye lens and the camera being so close to the face, results in a drastic difference in resolution between the upper and lower body (see Fig. 5.3). Consequently, estimating 2D or 3D pose from images captured from this first person viewpoint is considerably more challenging than from the more standard external perspective and therefore even state of the art approaches to human pose estimation [151] fail on our input data.

Our work tackles the two main challenges described above: *i)* given the unique visual appearance of our input images and the complete lack of training data for our specific scenario of a HMD mounted camera we have created a new photo-realistic synthetic dataset for training with both 2D and 3D annotations; and *(ii)* to tackle the challenging problem of self-occlusions and difference in resolution

Figure 5.2: Example images from our Ego-HMD Dataset showing the wide variety of characters, clothing, backgrounds and poses and the high quality of the renders. Our Ego-HMD Dataset contains a total of 383000 images and will be made publicly available.

between lower and upper body we have proposed a new architecture which takes into account uncertainty in the estimation of body joint positions.

More specifically, our solution adopts a two step approach. Instead of regressing directly the 3D pose from input images, we first train a model to extract the 2D heatmaps of the body joints and then regress the 3D pose via a dual-branch auto-encoder. The auto-encoder helps to hallucinate accurate joint poses for occluded body parts or those with high uncertainty. Both sub modules are first trained independently and finally end-to-end as the resulting network is fully differentiable.
The training is performed on real and synthetic data. The synthetic dataset was created with a large variety of body shapes, environments, and body motions, and will be made open access to the community for future research.

Finally, this proposed solution has deep impacts in human-robot interaction tasks since, unlike approaches that rely on single or multiple external cameras, it is more portable, it is able to reconstructs poses with a significant amount of occlusion and self-occlusion: something that external cameras cannot compare with.

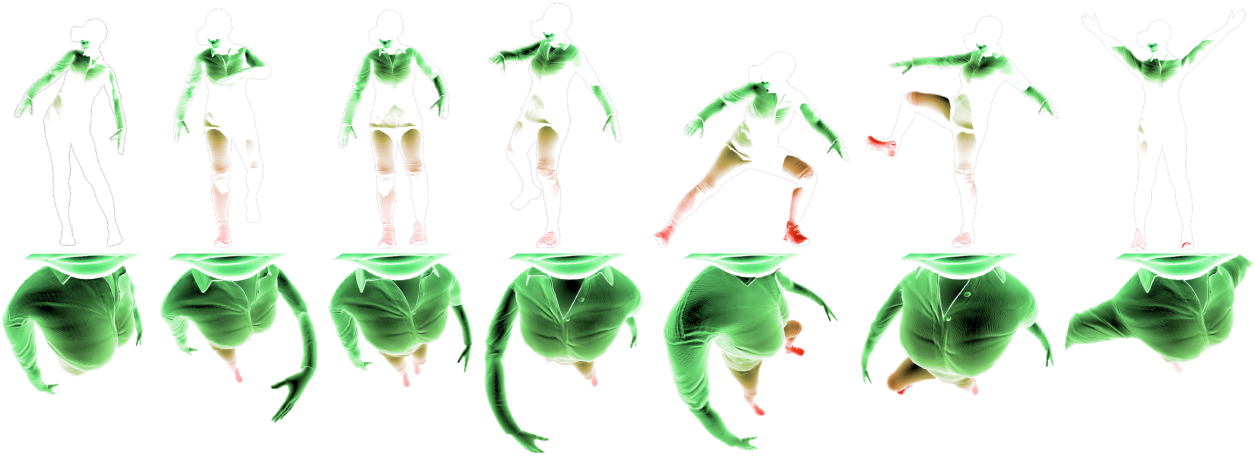


Figure 5.3: Visualization of different poses with the same synthetic actor.
Top: poses rendered from an external camera viewpoint. The blanked out body parts are those that would not be visible from the egocentric perspective.
Bottom: the same poses rendered from the egocentric camera viewpoint. The color gradient indicates the density of image pixels for each area of the body: *green* indicates higher pixel density, whereas *red* indicates lower density. This figure illustrates the most important challenges faced in egocentric human pose estimation: severe self-occlusions, extreme perspective effects and drastically lower pixel density for the lower body.

5.1.1 CONTRIBUTIONS

Novel modular egocentric encoder-decoder network for egocentric full-body 3D pose estimation from monocular images, captured from a camera equipped VR headset.The proposed approach firstly regresses 2D heatmaps that encode body joint positions, and then estimates the corresponding 3D joint locations via a novel multi-branch auto-encoder network. The main branch is responsible of regressing 3D poses from the pose embedding, while the additional auxiliary branches reconstruct information that helps to shape the latent space. The redundancy in this information enforces the latent vector to encode the uncertainty of the 2D joint estimates, as well as to preserve the limb orientation information. Both modules are first trained independently and finally end-to-end as the resulting network is fully differentiable.

Versatile 3D pose representation: the modular design of our proposed multi-branch auto-encoder architecture allows us to easily change the representation of the pose depending on the task at hand: from 3D joint location estimation to lo-

cal joint rotations that can be used to drive a virtual character based on the user's movements.

Synthetic dataset: a unique photo-realistic large-scale training corpus, composed of $383K$ frames rendered from a novel viewpoint (a fish-eye camera mounted on a VR display). It has superior photo-realism and a larger variability in the data with respect to the only other available monocular egocentric dataset Mo^2Cap^2 [124]. This dataset is already publicly available to promote progress in the area of egocentric human pose capture.

Performance analysis for egocentric pose estimation: analysis of different well-known 2D pose estimators combined with the proposed multi-branch auto-encoder architecture to measure their performance under different conditions. Furthermore, we explore how the architectures can be tailored for better pose reconstructions when using both synthetic and in-the-wild input images.
We conducted quantitative and qualitative evaluations on both synthetic and real-world benchmarks with ground truth 3D annotations, showing that our approach outperforms previous egocentric state-of-the-art Mo^2Cap^2 [124] by more than 25%. In addition, we achieve state-of-the-art performance on the more standard front-facing cameras 3D human pose reconstruction scenario, without any architecture modifications, performing second best after [152] on the Human3.6M benchmark [129].

5.2 3D POSE DETECTION FRAMEWORK

The problem of egocentric human pose estimation differs considerably from normal 3D human pose estimation approaches for the following reasons:

- Standard approaches assume *front facing cameras*. The person is most of the time entirely captured and all parts of the body have a uniform pixel density whereas in the egocentric scenario, the various parts of the body are captured with quite different coverage areas in the image and not always within the field of view (see Fig. 5.3).

- Front-facing cameras capture bodies with little self-occlusion. On the contrary, form the egocentric perspective the lower-body areas are self-occluded most of the time making the problem much harder to solve and requiring a strong inference machine to deal with the problem. An example of the severity of this problem is shown as white areas in the top row of Fig. 5.3.

- Standard approaches usually use pinhole camera models with relatively low lens distortion. The subject usually also stands at a relatively large distance allowing to even assume orthogonal projection. Instead, egocentric approaches use wide or fisheye lenses to capture as much of the body as possible to increase the observability of the body parts (see bottom row of Figure 5.3).

To tackle the problem, the proposed solution involves a deep neural network with two main sub-modules: *a)* a 2D-heatmap regressor, and *b)* a novel dual-branch auto-encoder that outputs 3D poses, using the predicted heatmaps as input, while enforcing the latent vector to encode the uncertainty information in the body joints. We show that this strategy not only leads to improved accuracy in the 3D pose estimates (particularly in the lower body) but also to better generalization to real data. Due to the complete lack of visual data with ground truth for training and evaluation, we present the Ego-HMD Dataset, a new large-scale photo-realistic open-access synthetic dataset.

Our experimental results demonstrate that our novel architecture provides substantial improvements over a 2D-to-3D state-of-the art and in-the-wild input images monocular human pose estimation approach with good qualitative results on real data captured from a camera installed on a VR headset $2.1cm$ away from the face.

In the next sections each of the individual modules of the proposed architecture will be described in details.

5.2.1 ARCHITECTURE

Our proposed architecture, shown in Fig. 5.4, is a two step approach with two modules. The first module detects 2D heatmaps of the locations of the body joints in image space using a ResNet [153] architecture. The second module takes the 2D heatmaps as inputs and regresses the 3D coordinates of the body joints using a novel dual branch auto-encoder.

The most important advantage of this pipeline approach, which decouples 2D and 3D estimation, is that each module can be trained independently, according to the available training data. For instance, if a sufficiently large corpus of images with 3D annotations is unavailable, the 3D lifting module can be trained using 3D mocap data and projected heatmaps without the need of paired images. Once the two modules are pre-trained the entire architecture can be fine-tuned end-to-end since it is fully differentiable.

5.2.2 2D POSE DETECTION

Given an RGB image $I \in \mathbb{R}^{368\times368\times3}$ as input, the 2D pose detector has the purpose of identifying 2D poses, represented as a set of heatmaps $HM \in \mathbb{R}^{47\times47\times15}$, one for each of the body joints.

For this task we have used a standard *ResNet 101* [153] architecture, where the last average pooling and fully connected layers have been replaced by a deconvolutional layer, with kernel size $= 3$ and stride $= 2$. The weights have been randomly initial-

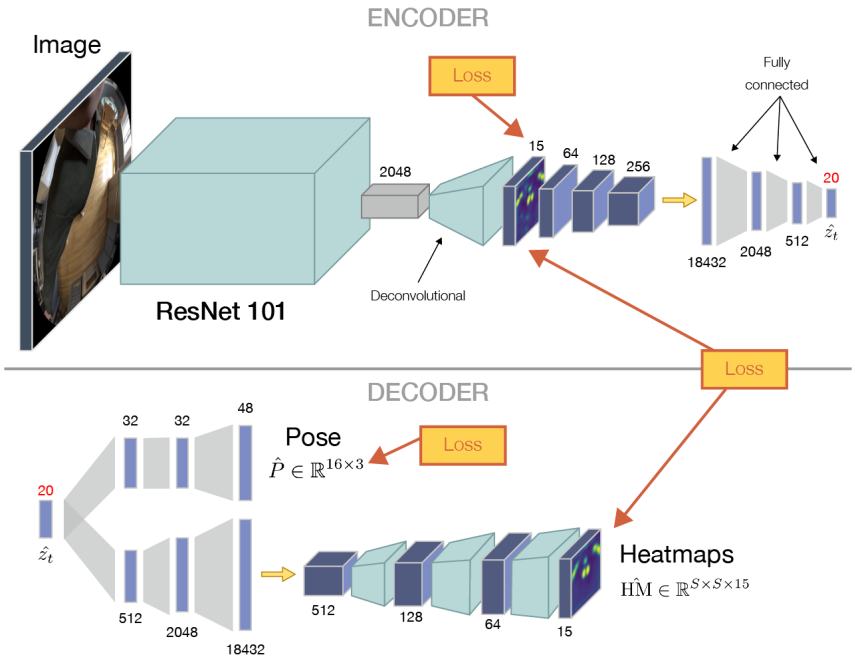


Figure 5.4: Our novel two-step architecture for egocentric 3D human pose estimation has two modules: *a)* the 2D heatmap estimator, based on ResNet101 [153] as the core architecture; *b)* the 3D lifting module takes 2D heatmaps as input and is based on our novel dual branch auto-encoder.

ized using Xavier initialization [154]. The model was trained using normalized input images, obtained by subtracting the mean value and dividing by the standard deviation, and using the MSE of the difference between the ground truth heatmaps and the predicted ones as the loss:

$$Loss_{2\mathrm{D}} = mse(HM, \hat{HM}) \tag{5.1}$$

5.2.3 2D-TO-3D MAPPING

The 3D pose module takes as input the 15 heatmaps computed by the previous module and identifies the final $3D$ pose $P \in \mathbb{R}^{16\times3}$. Note that the number of output 3D joints is 16 since we include the head, whose position cannot be estimated in the $2D$ images since the person is wearing a headset; but can be regressed in $3D$.

In most pipeline approaches the *3D lifting* module typically takes as input the 2D

coordinates of the detected joints. Instead, similarly to [155], our approach regresses the 3D pose from heatmaps, not just 2D locations. The main implication is that the heatmaps carry important information about the uncertainty of the 2D pose estimates.

The main novelty of our architecture (see Fig. 5.4), and where it departs from other human pose estimation models, is that we ensure that this uncertainty information is not lost. We use an auto-encoder with a dual-branch decoder that ensures the latent vector encodes information about the uncertainty in the joints. While the encoder takes as input a set of heatmaps, and encodes them into a low-dimensional vector $\hat{z}$, the decoder has two branches — one that regresses the 3D pose from $\hat{z}$ and another that reconstructs the input heatmaps. The purpose of this branch is to force to map back the latent vector into the probability density function of the solution found by the 2D regressor, e.g., the heatmaps.

The overall loss function for the auto-encoder becomes

$$Loss_{\text{AE}} = ||P - \hat{P}||^2 + \lambda_{hm}||\hat{HM} - \hat{\hat{HM}}||^2 + \lambda_c R(P, \hat{P}) \tag{5.2}$$

where $\hat{P}$ is the pose predicted by the decoder and P the ground truth; $\hat{\hat{HM}}$ is the set of heatmaps regressed by the decoder from the latent space and $\hat{HM}$ are the heatmaps regressed by ResNet (see Sec. 5.2.2). Finally R is an additional loss operating on the poses only $R(P,\hat{P}) = \lambda_\theta \theta(P,\hat{P}) + \lambda_L L(P,\hat{P})$ with

$$\theta(P,\hat{P}) = \sum_{l}^{L} \frac{P_l \cdot \hat{P}_l}{||P|| * ||\hat{P}_l||}$$
$$L(P,\hat{P}) = \sum_{l}^{L} ||P_l - \hat{P}_l||$$

corresponding to the cosine-similarity error and the limb-length error, with $P_l \in \mathbb{R}^3$ the l^{th} limb of the pose.

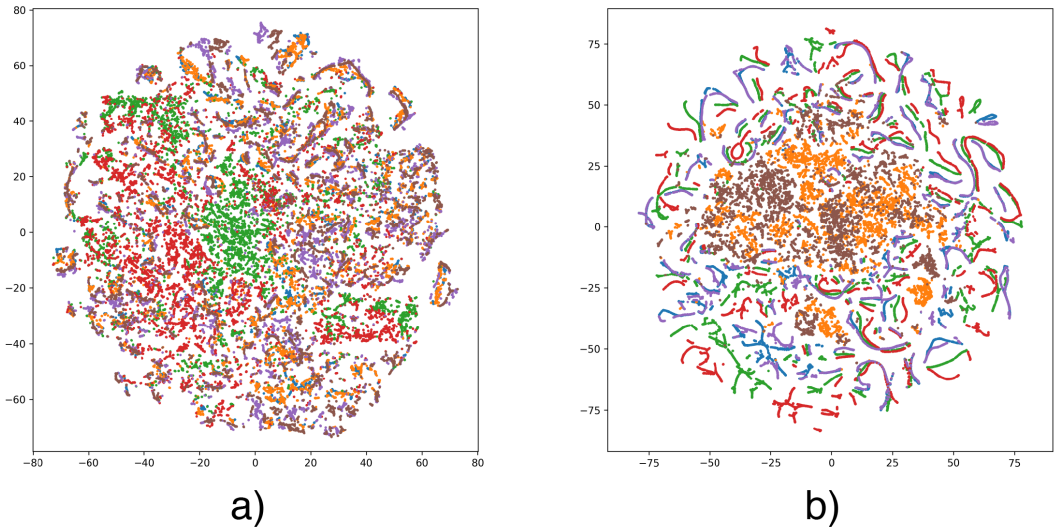


Figure 5.5: t-SNE plot showing projections of the training data mapped on the latent space with each color representing a different character, when: *a)* the auto-encoder consists of a single decoder; or *b)* the decoder has a second branch as regularizer used only during training. The dual-branch architecture generates a much better distributed latent space that solves generalization issues.

Single vs. double branch decoder: Visualizing the reconstructed poses, it is clear that the single-branch approach is failing mainly in reconstructing the lower body, failing to detect the identity of the actors (set of limb lengths), and as a consequence also the position of the lower joints.

To better understand the effect of the second branch in the auto-encoder, a better and more in depth analysis over the latent space has to be performed to understand if the better results comes from an expected behaviour.

Figure 5.5 shows further justification for our proposed dual-branch decoder. The plots show the distribution of training poses after mapping them into the latent space and projecting in 2D using t-SNE, in two cases: *a)* when the AE was trained with a single-branch encoder, or *b)* when the AE was trained with the proposed dual-branch architecture. Here the visualization of the distributions per character: each color represent a different character.

Quantitative results of the ablation study are shown in Sec. 5.4.2.

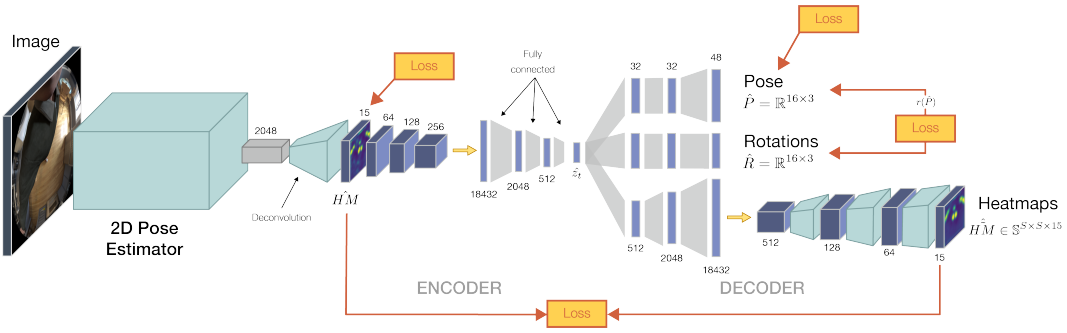


Figure 5.6: Egocentric pose estimation architecture extended to include rotation pose representation. A new decoder branch is introduced to be able to change the pose representation without the need to re-train the model and to guarantee even better smoothness of the learned latent space.

It is clear that forcing the latent space to learn to embed the uncertainty of the 2D estimates, results in a much better distributed space, leading to better generalization than the single-branch architecture (as numerically demonstrated in Sec. 5.4.2) where we found that the average 3D error in the pose reconstructions is more than halved when the second branch is added to the decoder.

When looking at Image *(b)* of Figure 5.5 some "string-like" shapes can be seen in the distribution. After inspection (see results in Sec 5.4.7) of the latent space with the related poses, it is possible to see that they corresponds to poses in which the person is staying in a steady pose and only one of the limbs is moved.

5.2.4 POSE ROTATION REPRESENTATION

The architecture definition visualized in Fig. 5.4 can be slightly modified to extend the capabilities of the model in representing poses, as shown in Fig. 5.6. Noticeably, the architecture is very similar with the only difference of a third branch in the decoder section where the rotation representation is now an additional output.

Due to this new addition, the loss is updated as follows:

$$\begin{aligned} \mathrm{L_{AE}} &= \lambda_p(||\mathbf{P}-\hat{\mathbf{P}}||^2 + W(\mathbf{P},\hat{\mathbf{P}})) + \\ &\quad \lambda_r||FK(\hat{\mathbf{R}})-\hat{\mathbf{P}})||^2 + \\ &\quad \lambda_{hm}||\widehat{\mathbf{HM}}-\widetilde{\mathbf{HM}}||^2 \end{aligned} \tag{5.3}$$

with $\mathbf{P}$ the ground truth; $\hat{\mathbf{R}}$ the predicted local joint rotations and $FK(\hat{\mathbf{R}})$ the function that estimates joint positions by performing differentiable forward kinematics using predicted rotations; $\widetilde{\mathbf{HM}}$ is the set of heatmaps regressed by the decoder from the latent space and $\widehat{\mathbf{HM}}$ are the heatmaps regressed by 2D pose estimator module. Different local joint rotation representations were tested and ultimately a Quaternion representation was chosen due to the stability of the rotations during training, leading to more robust models. The rotation branch also helps generating better results as shown in Sec. 5.4.2 with smoother transitions on consecutive frames on poses estimated frame-by-frame.
All the losses and regularizers previously defined and not specified in this section remain the same.

5.2.5 IMPLEMENTATION DETAILS

The model has been trained on the entire train-set of our custom dataset described in Sec 5.3 for 3 epochs, with a learning rate of $1e-3$ using batch normalization on a mini-batch of size 16. The deconvolutional layer used to identify the heatmaps from the features computed by *ResNet* has kernel size $= 3$ and stride $= 2$. The convolutional and deconvolutional layers of the encoder have kernel size $= 4$ and stride $= 2$. Finally, all the layers of the encoder use leakly ReLU as activation function with 0.2 leakiness.

The λ weights used in the loss function were identified through grid search and set to $\lambda_{hm} = 10^{-3}$, $\lambda_c = 10^{-1}$, $\lambda_\theta = 10^{-2}$ and $\lambda_L = 0.5$.
Finally, the model has been trained from scratch with Xavier weight initializer.

5.3 DATASET

Data driven applications like Deep Learning approaches require a large amount of data to work with. The larger the dataset the better the model can learn how to perform a specific task.

All available datasets for 3D Human Pose Estimation have been generating under the condition that the camera/all the cameras are statically placed in an environment pointing towards the person, such that the person contained in the images is most of the time entirely captured and all body parts have a uniform pixel density.
To better understand this, Fig 5.7 shows the camera placement as well as an example of frame contained in the Human3.6m dataset described in the previous chapters (see Sec 3.3 and Sec 4.3).

5.3.1 *X*R-EGOPOSE DATASET

In the design of the dataset, we were focused on scalability, with augmentation of characters, environments, and lighting conditions. A rendered scene is randomly generated from the total number of characters, environments, lighting rigs, and animation actions that are then re-targeted from mocap data. The headset on which the camera is mounted can be randomly offset and re-positioned for each character during this generation process. Figure 5.8 shows some sampled characters taken

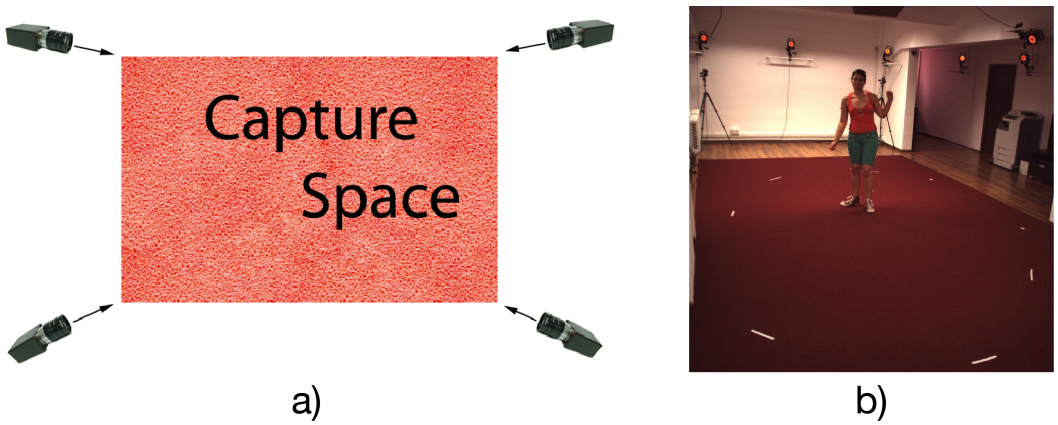


Figure 5.7: Example of dataset assuming front facing cameras: *a)* camera placement; *b)* image capture from one of those cameras.

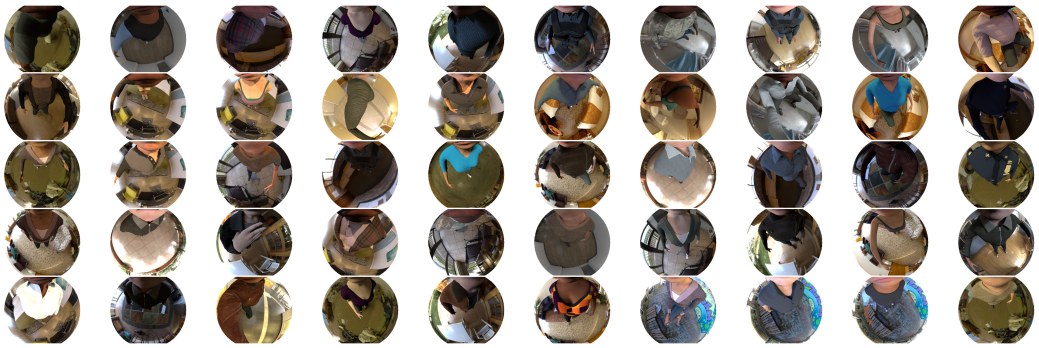


Figure 5.8: Sample of characters generated with randomly selected scenes and lighting conditions from the xR-EgoPose dataset. Notice the data diversity in terms of skin color, illumination, clothing, environments and actions.

from the generated dataset.

During character creation, from a single average character we generate additional *a)* skinny short, *b)* skinny tall, *c)* full short and *d)* full tall different versions. The height distribution of each of the characters can be seen in Fig 5.9. This is to improve the diversity of body types and occlusion areas from a single camera perspective.

Skin: we started with a diverse set of skin colors from the initial set of scanned actors. Color tones include *white* (Caucasian, freckles or Albino), *light-skinned European*, *dark-skinned European* (darker Caucasian, European mix), *Mediterranean or olive* (Mediterranean, Asian, Hispanic, Native American), *dark brown* (Afro-American, Middle Eastern) and *black* (Afro-American, African, Middle Eastern). We also built random skin tone parameters into the shaders of each character used with the scene generator.

Clothing: we focused on silhouettes, textures, local colors. Clothing types (modeled by an artist) include Athletic Pants, Jeans, Shorts, Dress Pants, Skirts, Jackets, T-Shirts, Long Sleeves, Tank Tops. Various shoes including, Sandals, Boots, Dress Shoes, Athletic Shoes, Crocs.

Actions: the characters included in the dataset perform a variety of actions that

have been clustered to nine broad classes: *Gaming*, *Gesticulating*, *Greeting*, *Lower (body) stretching*, *Patting*, *Reacting*, *Talking*, *Upper (body) stretching* and *Walking*.

Images: we rendered a total of 383 thousand 1024×1024 fisheye 16-bit images sampled at a rate of 30fps. *Rgb*, *depth*, *normals*, *body segmentation*, and *pixel world position* images are generated for each frame, with the option for exposure control for augmentation of lighting.

A *json* file is provided for each frame including joint positions in world space, height of the character, environment, camera pose, body segmentation id, and animation rig.

In addition to the render scene generator we built a post composite process for adding 3D decals, which consists into a material projected onto other geometry used to add variety to a scene. This means we can perform this operation to already rendered scenes for clothing augmentation without having to re-render the entire scene and decreasing the render time by a factor of 100.

Overall quality of the data was optimized for render times targeted under 30s on a Titan X GPU, or 130s 12 core CPU per frame. The characters were created initially with global custom shader settings applied across clothing, skin, and lighting of environments for all rendered scenes. This was to keep things normalized and to maintain a physically based rendering setup.

5.3.1.1 Properties

The dataset has a total size of 383 thousand frames, with 23 male and 23 female characters, divided into three sets: *Train-set* with size $252K$ frames; *Test-set* with size $115K$ frames; and the *Validation-set* of size $16K$ frames.

The gender distribution in the sets is the following one: 13, 7 and 3 males and 11,

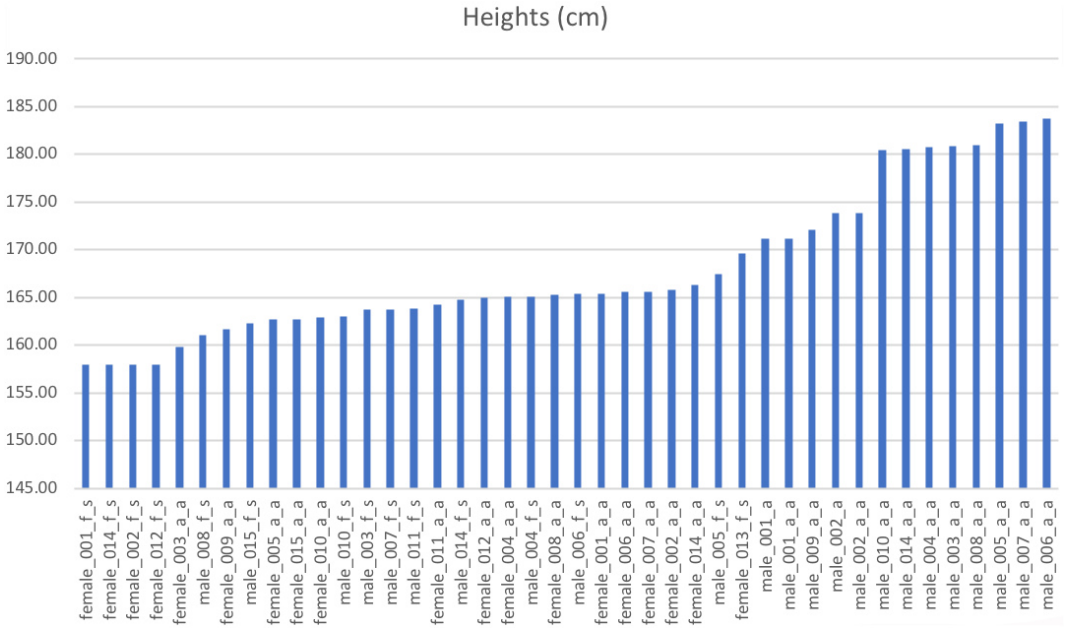


Figure 5.9: Distribution of heights in the dataset for the different actors; sorted from shortest to tallest.

Action	N. Frames	Size Train	Size Test
Gaming	24019	11153	4684
Gesticulating	21411	9866	4206
Greeting	8966	4188	1739
Lower Stretching	82541	66165	43491
Patting	9615	4404	1898
Reacting	26629	12599	5104
Talking	13685	6215	2723
Upper Stretching	162193	114446	46468
Walking	34989	24603	9971

Table 5.1: Total number of frames per action and their distribution between train and test sets.

5, and 3 females respectively for train, test and validation set.

Table 5.1 provides a detailed description about how the dataset has been partitioned according to the different actions.

Both quantitative and qualitative results are shown in Sec. 5.4.

5.3.2 *X*R-EGOPOSE^R DATASET

The $\sim$ 10K frames of our small scale real-world data set were captured from a fish-eye camera mounted on a VR HMD worn by three different actors wearing different clothes, and performing 6 different actions. The ground truth 3D poses were acquired using an internal custom mocap system. The network was trained on our synthetic corpus (*x*R-EgoPose) and fine-tuned using the data from two of the actors. The test set contained data from the unseen third actor.

5.3.3 EGOCAP DATASET

A competitor egocentric dataset working in different conditions than our has been introduced by Rhodin *et al.* [122] called EgoCap.

This is a realistic dataset annotated with the joint locations of a kinematic skeleton and other body parts such as the hands and feet, whose camera setup is shown in Figure 5.10.

To avoid the tedious and error-prone manual annotation of locations in thousands of images, the authors use a state-of-the-art marker-less motion capture system (Captury Studio of The Captury) to estimate the skeleton motion in 3D from eight stationary cameras placed around the scene. The skeleton joints are then projected into

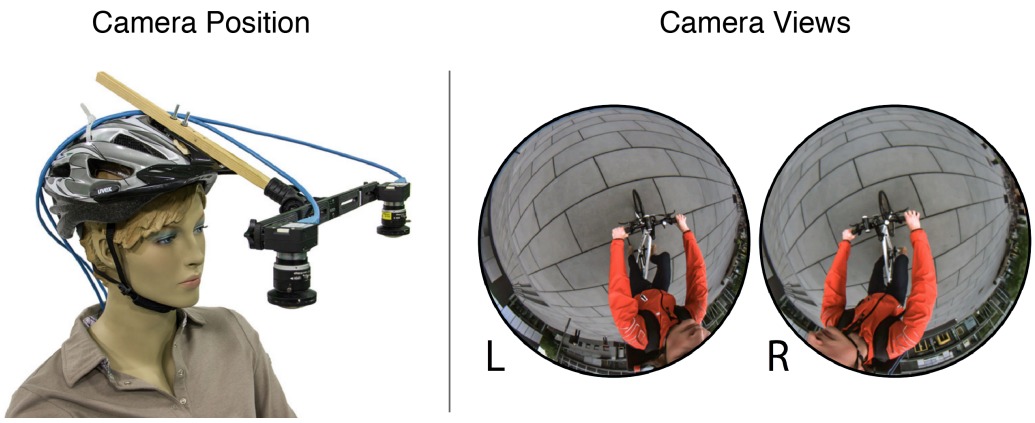


Figure 5.10: *Left)* EgoCap [122] dataset camera setup. A different camera configuration has been mounted on Oculus goggles in a similar setup; *Right)* Camera perspective sample.

the fisheye images of their head-mounted camera rig.

The authors mention that several videos were recorded capturing eight subjects performing various motions in a green-screen studio. For the training set, chroma keying the background of each video frame has been replaced with a random, floor-related image downloaded from Flickr.

In addition, data augmentation is performed by changing the appearance of subjects, varying the colors of clothing, while preserving shading effects, using intrinsic re-coloring.

The resulting dataset consists of a train-set containing approximately 75.000 six-subjects annotated fisheye images. Two additional subjects are captured and prepared for validation purposes.

5.3.3.1 Comparison with xR-EgoPose dataset

Due to the nature of this dataset, it is natural to ask the set of differences that distinguish this dataset from the former one (xR-EgoPose dataset).
Unlike this dataset, our xR-EgoPose dataset is monocular only, for which the 3D pose inference is performed using a single camera. Furthermore, the distance of the camera to the character face is dramatically different making the problem way harder to solve, due to the increasing about of body self-occlusion.

A good visual interpretation of this is shown in Figure 5.11, where a side-by-side comparison of the camera point of view between these two dataset is highlighted.

5.3.4 MO2CAP2 DATASET

A competitor egocentric dataset has been introduced in Mo2Cap2 [124], which has been built on top of the large scale synthetic human SURREAL dataset [143], designed specifically for their approach where a camera is mounted on a baseball cap,

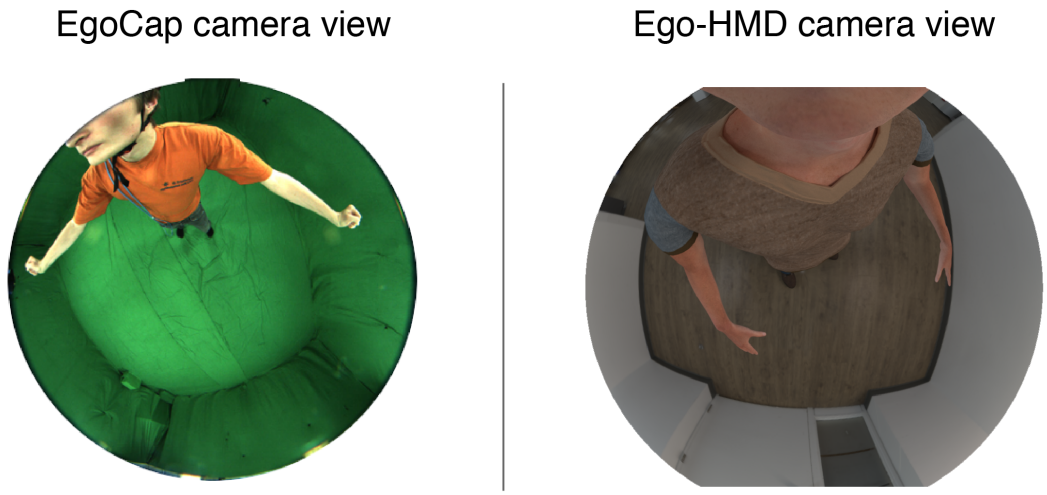


Figure 5.11: Camera perspective comparison between EgoCap and xR-EgoPose datasets. Our dataset camera position generates more body self-occlusions which results in more challenging predictions.

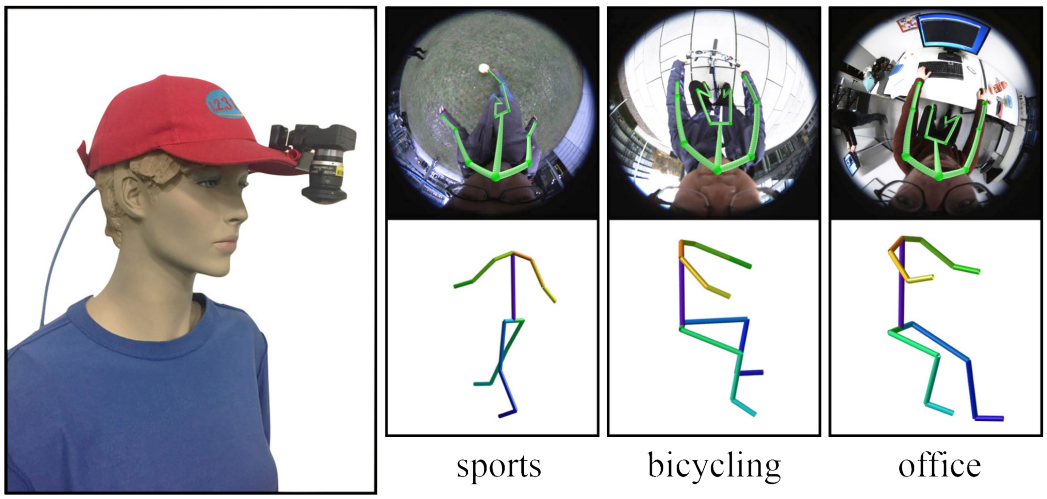


Figure 5.12: Mo2Cap2 [124] egocentric approach camera setup.

approximately 8 cm away from the nose, see Figure 5.12. Such camera configuration results in a lower level of self-occlusion due to the larger distance of the camera to the nose.

They animate characters using the SMPL body model [88] with uniformly sampled motions from the CMU MoCap dataset [156]. Body textures are chosen randomly from the texture set provided by the SURREAL dataset [143].

A total of 530,000 images were rendered, which encompass around 3000 different actions and more than 700 different body textures. Additionally, for improving realism, there is a process which tries to mimic the camera according to both the light condition and the background of the captured real world. Some sample frames are shown in Figure 5.13.

Figure 5.13: Mo2Cap2 [124] sampled frames from the synthetically generated dataset.

5.3.4.1 Details

Images are rendered from a virtual fisheye camera attached to the forehead of the character at a distance similar to the size of the brim of the used real world baseball cap. To this end, they calibrate the real world fisheye camera using the omnidirectional camera calibration toolbox ocamcalib and apply the intrinsic calibration parameters to the virtual camera.

The characters are rendered using a custom shader that models the camera distortion of a fisheye camera. Note that the camera position with respect to the head might

change slightly, due to the camera movements and varying wearing angles and positions of the cap. To simulate this effect, the authors add random perturbation to the virtual fisheye camera position.
Finally, random spherical harmonics illumination is used with a special parameterization to ensure a realistic top down illumination.

All images are augmented with the backgrounds chosen randomly from a set of more than 5000 indoor and outdoor ground plane images captured by our fisheye camera. To gather such background images, a fisheye camera has been attached to a long stick, in such a way of obtaining images that do not show the person holding the camera.

5.3.4.2 Comparison with xR-EgoPose dataset

Due to the nature of this dataset, it is natural to ask the set of differences that distinguish this dataset from the former one (xR-EgoPose dataset).

Apart from the obvious different camera placement, which is a non-trivial difference that consequently makes the problems harder to solve mainly due to the larger amount of self-occlusion, both datasets consists of a set of synthetically generated images. The difference is that the former dataset renders scenes in which the character is part of it, whereas the latter only renders the character which is then added to the image and adjusted according to the environment. The solution implemented in our dataset (Ego-HMD) has the **significant effect** of generating shadows both in the environment and on the character that can greatly affect the difficulty of the task. This statement can be quantitatively verified in the Experimental section 5.4.

To better illustrate this, see Figure 5.14, where we show two different frames representing a single character wearing the same clothes, placed in the same environment, and as it is possible to see hot the shadow changes. Since the second pose is facing the right part of the body towards the window, the left part becomes more

challenging to correctly predict. These kind of effects is what is needed in order to have a real life application that works under all different conditions.

Finally, the last and most important characteristic that differentiates the two datasets is the quality of the synthetic images. A side by side comparison is shown in Figure 5.15.

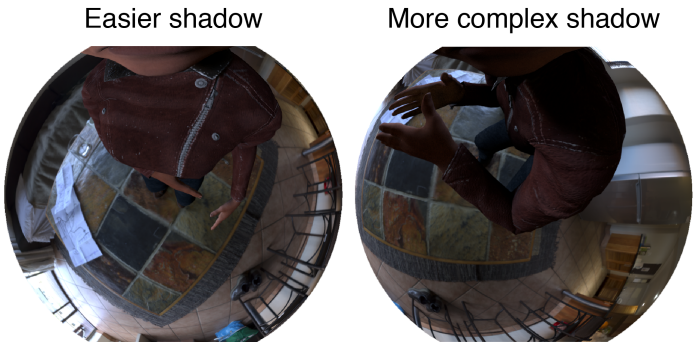


Figure 5.14: Shadow comparison between two poses with a character wearing the same clothes in the same environment.

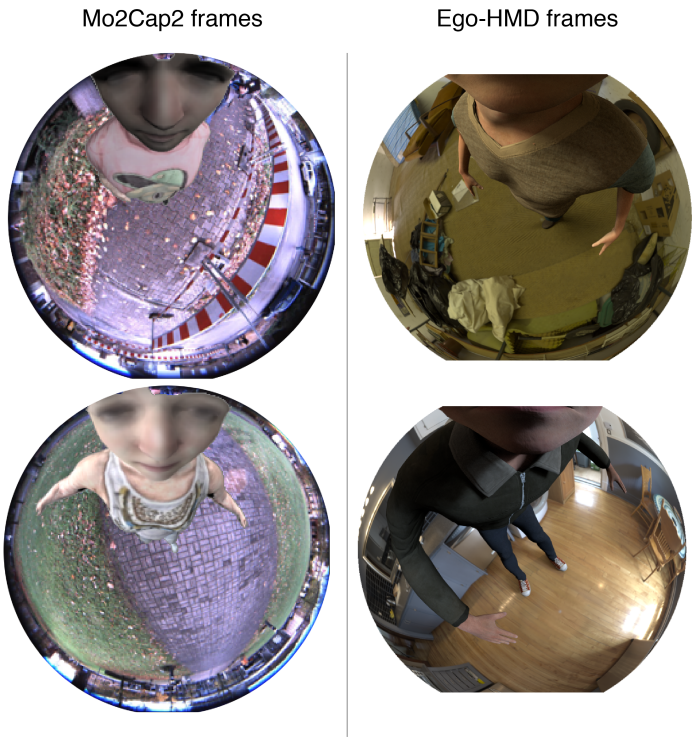


Figure 5.15: Sample frames from both dataset to show the difference in quality of the synthetic images.

5.4 EXPERIMENTAL EVALUATION

In the following, we thoroughly evaluate our proposed approach on our novel *x*R-EgoPose dataset, we perform parameter and architecture ablations, and we evaluate on the real-world Mo^{2}Cap^2 test-set [157] which includes 2.7K frames of real images with ground truth 3D poses of two people captured in indoor and outdoor scenes. In addition, we show qualitative results on our controlled small-scale real-world dataset and demonstrate how our approach can be used to animate virtual characters for *x*R telepresence.

Finally, we evaluate quantitatively on the Human3.6M dataset to show that our architecture generalizes well without any modifications to the case of an external camera viewpoint 3D human pose estimation with state-of-the-art results.

5.4.1 EVALUATION PROTOCOL

Taking inspiration from the evaluation protocol defined in the Human3.6M dataset (see Sec 3.3 and Sec 4.3), we use as an evaluation protocol to asses the quality of the reconstructions by sampling every frame of the test-set using as an error function

$$E(P,\hat{P}) = \frac{1}{N_f}\frac{1}{N_j}\sum_{f=1}^{N_f}\sum_{j=1}^{N_j}||P_j^{(f)} - \hat{P}_j^{(f)}||_2 \tag{5.4}$$

where $P_j^{(f)}$ and $\hat{P}_j^{(f)}$ are the 3D locations for the ground truth pose and the predicted pose at frame f for joint j. This is averaged over the N_f number of frames and N_j number of joints.

To ensure high reproducibility of our results on our novel synthetic x R-EgoPose dataset, we first evaluate our method on a randomly initialized ResNet 50. We intentionally do not perform any pre-training strategies as shown later in this section. This affects the final results. We want to establish the x R-EgoPose as a benchmark dataset and therefore report reproducible numbers, that have been computed using a standard network architecture, trained with a simple protocol.

5.4.2 QUANTITATIVE RESULTS ON XR-EGOPOSE

Firstly, we evaluate our approach on the test-set of our synthetic *x*R-EgoPose dataset. Unfortunately, it was not possible to establish a comparison on our dataset with state-of-the-art monocular egocentric human pose estimation methods such as Mo^2Cap^2 [157] given that their code has not been made publicly available. Instead we compare with Martinez *et al.* [82], a recent state of the art method for a traditional external camera viewpoint.

Such method is a pipeline-approach, in which two – sometimes independent – modules are used sequentially: the output of the former is used as input of the latter. Here, the second module is given 2D joint locations on the image, and it predicts the 3D positions of the joints.

For a fair comparison, the training-set of our *x*R-EgoPose dataset has been used to re-train the model of Martinez *et al.*; This way we can directly compare the performance of the 2D to 3D modules.

There are several state-of-the-art approaches other than Martinez's which employs

Approach	Gaming	Gesticulating	Greeting	Lower Stretch.	Patting	Reacting	Talking	Upper Stretch.	Walking	All (mm)
Martinez [82]	109.6	105.4	119.3	125.8	93.0	119.7	111.1	124.5	130.5	122.1
Ours — p3d	138.3	108.5	100.3	133.3	117.8	175.6	93.5	129.0	131.9	130.4
Ours — p3d+rot	110.7	90.9	91.9	119.1	98.6	106.8	86.9	88.0	88.2	91.2
Ours — p3d+hm	56.0	50.2	44.6	51.1	59.4	60.8	43.9	53.9	57.7	58.2
Ours — p3d+hm+rot	60.4	54.6	44.7	56.5	57.7	52.7	56.4	53.6	55.4	**54.7**

Table 5.2: Quantitative evaluation with Martinez *et al.* [82], a state-of-the-art approach developed for front-facing cameras. Both upper and lower body reconstructions are shown as well. A comparison with our own architecture where different configurations are analyzed. Specifically, the impact of the additional branches is evaluated. Note how the competing approach fails consistently across different actions in lower body reconstructions. This experiment emphasizes how, even a state-of-the-art 3D lifting method developed for external cameras fails on this challenging task. It also emphasizes the contribution of encoding uncertainty for achieving low-reconstruction errors.

a pipeline strategy, like the one proposed by Zhou *et al.* [73], and their main drawback consists in not fully exploiting the information generated by the 2D pose detector when regressing the final 3D pose. Namely, the uncertainty: such approaches usually perform an *argmax* operation over the generated heatmaps, loosing the uncertainty of the 2D estimations and being left only with their (u, v) position.

Using the evaluation protocol defined in 5.4, we compare the reconstruction error of our proposed approach with the one by Martinez*et al.* [82], showing the results in Table 5.2. Our approach outperforms the one proposed by Martinez *et al.* by 36.4% in the upper body reconstruction, 60% in the lower body reconstruction, and 52.3% overall, showing a considerable improvement.
This emphasizes the complexity of the task as well as how important is to properly identify the identity of the person when retrieving its pose. A failure in doing this leads to a drastically larger reconstruction error for the lower body, which is the part of the body relying more on prior knowledge to infer joint positions, due to the large amount of self-occlusions.

5.4.2.1 Effect of the decoder branches

Table 5.2 reports an ablation study to compare the performance of three versions of our approach. We report results using: *i)* only 3D pose supervision only (Ours — p3d); *ii)* additional supervision on regressed rotations (Ours — p3d+rot); *iii)* and on regressed heatmaps (Ours – p3d+hm); finally for our novel *multi-branch* auto-encoder supervised on all three signals (Ours — p3d+hm+rot).

The overall average error of the single branch encoder is 130.4 mm, far from the 54.7 mm error achieved by our novel *multi-branch* architecture. The dual branch encoders produce an error of 91.2 mm and 58.2 mm, respectively. Ours results clearly demonstrate that all branches contribute to our final result. Both, forcing the network to encode uncertainty of the 2D joint estimates by regressing heatmaps, as well as preserving the limb orientation information by regressing rotations, helps to

estimate better 3D poses.

5.4.2.2 Reconstruction error per joint type

Table 5.3 reports a decomposition of the reconstruction error into different individual joint types. The highest errors are in the hands and feet. This observation is in accordance with the fact that hands and feet are often not or only barely visible. Hands can go out of the camera field of view e.g. by lifting or stretching the arms or may be occluded by the body. Feet are only visible when the subject looks slightly down and always cover only a very small portion of the image, due to the strong distortion. Nevertheless, our method always predicts plausible poses, even for high occlusions as displayed in Fig. 5.22.

Additional statistical analysis over the joint reconstruction errors over the test-set is included in Figure 5.16, by presenting the log-probability errors of the model on our novel *x*R-EgoPose dataset. As already mentioned, this test confirms as well that the neck is the easiest joint to detect, whereas the hands are the worse performing joints.

We find the joints contributing the most to the overall error are the hands. This is somehow in contrast with the larger lower-body reconstruction observed before-

Joint	**Error (mm)**	**Joint**	**Error (mm)**
Left Leg	34.33	Right Leg	33.85
Left Knee	62.57	Right Knee	61.36
Left Foot	70.08	Right Foot	68.17
Left Toe	76.43	Right Toe	71.94
Neck	6.57	Head	23.20
Left Arm	31.36	Right Arm	31.45
Left Elbow	60.89	Right Elbow	50.13
Left Hand	90.43	Right Hand	78.28

Table 5.3: Average reconstruction error per joint using Eq. 5.4, evaluated on the entire test-set (see Sec. 5.3) with model trained using only synthetic data.

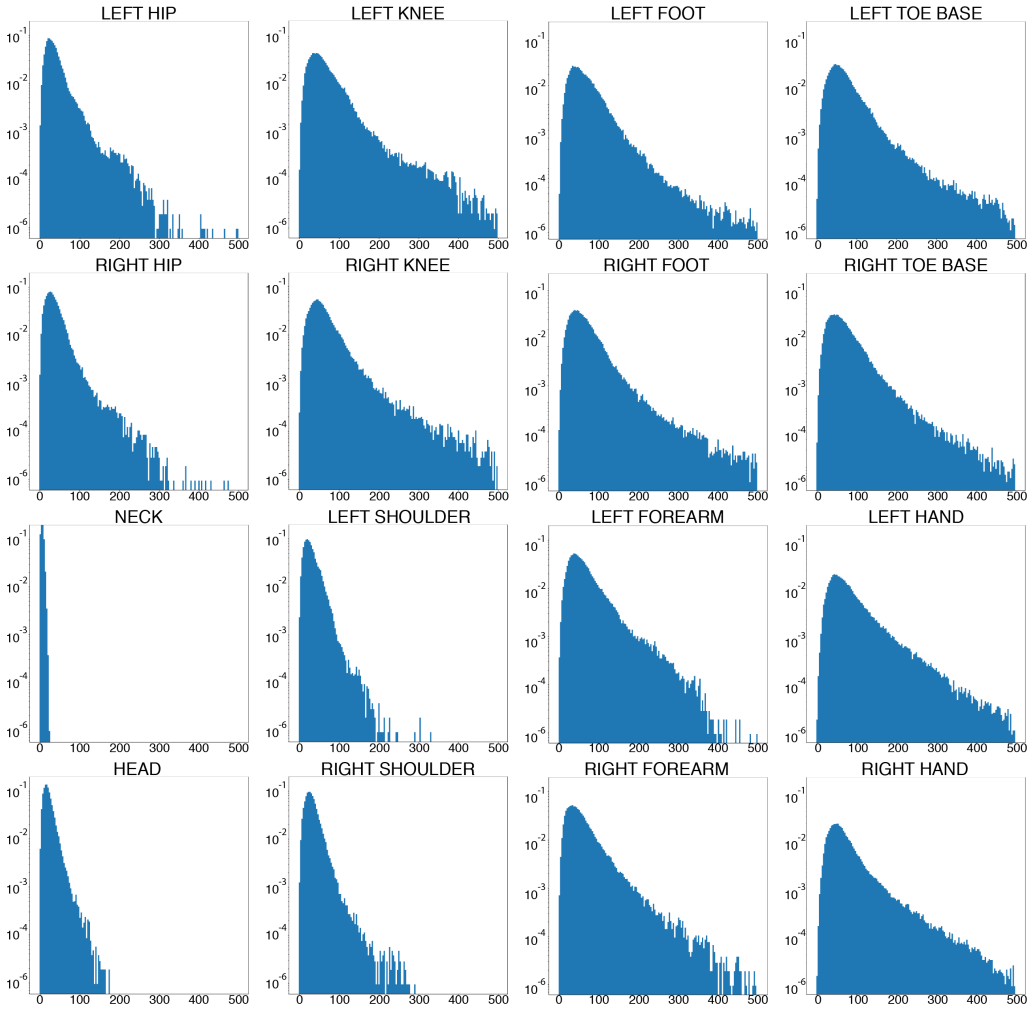


Figure 5.16: Analysis of the log-probability error of each joint evaluated on the *x*R-EgoPose test-set. The neck is the best performing joint whereas the hands are the worse performing ones.

hand. The main reasons for this to happen is that: *a)* the upper body reconstruction error includes neck and shoulders. Such joints are "more rigid" in the human anatomy and lower the overall average error; *b)* hands – and arms in general – correspond to the part of the human body that moves the most with a larger degree of freedom; this consequently means that is harder to learn the set of all possible motions from a normal size dataset; and *c)* hands might frequently fall outside the camera field of view. When this happens, as they are not observed, our approach fails to estimate their poses accurately: there are many possible interpretations – all correct – about the pose, since the hands are not accounted for the prediction. A

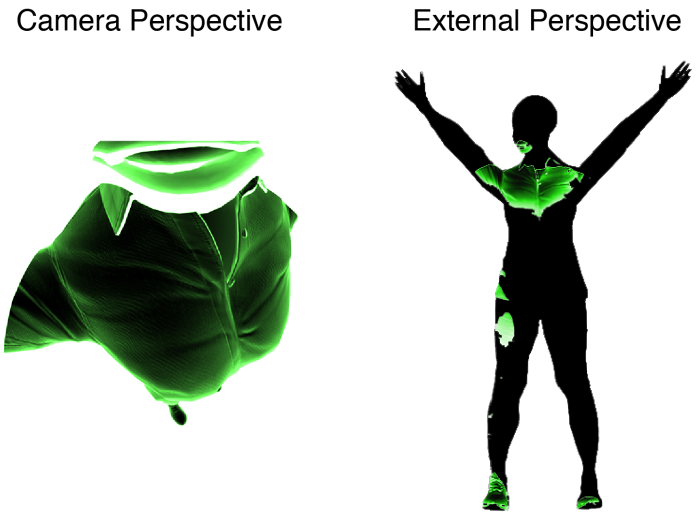


Figure 5.17: The area of the images highlighted in green represents what is visible from the egocentric camera perspective. In this particular pose, the hands are clearly outside the camera FoV (represented in black), and therefore it is not possible to precisely infer their position.

visual illustration of this particular situation is shown in Fig. 5.17.

5.4.3 ABLATION STUDIES

One of the main advantages of this proposed architecture is its structure, which allows to swap some of the modules for others for better customization over specific applications as well as to provide better generalization properties. In this section, we are performing ablation studies to further explore the capability of the approach.

5.4.3.1 Heatmap Estimation: Architecture Ablation

So far, we have used the established *ResNet 50* [153] architecture in all our experiments. In order to study the effect of the heatmap estimation network, we experiment with different architectures and initialization strategies. Specifically, we experiment with *ResNet 50* [153] and *U-Net* [158]. We use *ResNet 50* in two variants: randomly initialized using Xavier initialization [154] and pre-trained on ImageNet [159].

The *U-Net* is composed from a *ResNet 18* backbone encoder, pre-trained on ImageNet, and a randomly initialized decoder. The *ResNet 50* consists of 24.2 million

Configuration	Gaming	Gesticulating	Greeting	Lower Stretching	Patting	Reacting	Talking	Upper Stretching	Walking	All (mm)
ResNet 50	60.4	54.6	**44.7**	56.5	57.7	52.7	56.4	53.6	55.4	54.7
ResNet 50 (p)	**51.6**	**44.6**	64.6	52.4	**50.8**	**44.0**	46.5	51.4	52.8	51.1
U-Net (p)	52.5	49.2	72.0	**37.3**	53.0	44.4	**46.1**	**39.3**	**37.2**	**41.0**

Table 5.4: Performance analysis: different combinations of 2D pose detectors combined with the *multi-branch* lifting network. All variants have been trained and tested on the synthetic dataset. Variants with (p) have been pre-trained on ImageNet.

trainable parameters. The *U-Net* contains 18.3 million parameters. All variants produce the same heatmap resolution for better comparison. The lifting networks share the same architecture and number of parameters, but have been trained specifically for each 2D pose estimation network, to accommodate its unique heatmap properties. We additionally experimented with *ResNet 101* [153], *Convolutional Pose Machines* [15], and *Stacked Hourglass Network* [74]. These experiments resulted in comparable performance at a higher computational cost compared to *ResNet 50*, and are therefore not discussed further.

The experiments suggest that pre-training helps. The full pipeline using a pre-trained *ResNet 50* improves the MPJPE error to 51.1 mm, compared to 54.7 for random initialization, see Tab. 5.4. While a recent work [160] suggests that pre-training usually is not necessary, the authors describe two aspects where pre-training does help. First, pre-training helps faster convergence. Second, for small datasets, pre-training helps to improve accuracy. While our synthetic dataset is large, it features less variability in scenes and subjects, compared to large real-world datasets like e.g. MPII [161].

In a following step, we experiment using a *U-Net* for 2D pose estimation. Using a *U-Net* architecture boosts the performance of our pipeline and significantly improves the MPJPE error to 41.0 mm. Empirically, we found that the *U-Net*-based 2D pose estimator also generalizes, to a certain extent, to real data, predicting plausible heatmaps for unseen data, while only having been trained on our synthetic

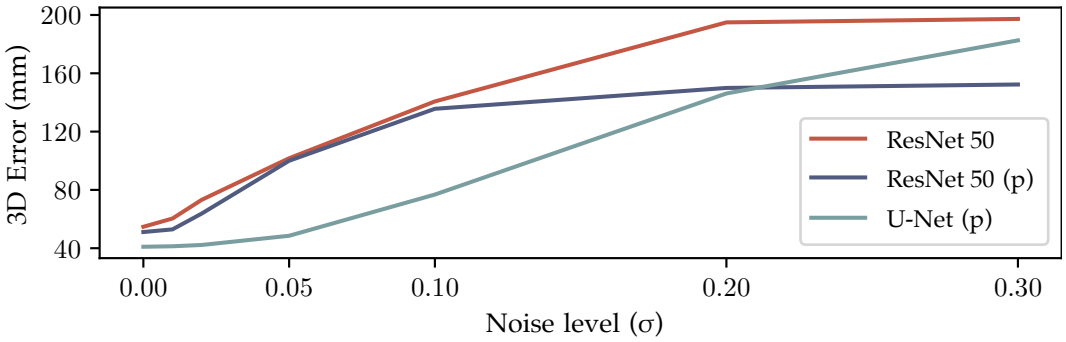


Figure 5.18: Performance of our proposed pipeline using different 2D pose estimation networks under the influence of white Gaussian noise in the image domain. Networks with (p) have been pretrained on ImageNet.

dataset.

The *Resnet 50*-based estimator fails without prior refinement. We hypothesize, that the improved performance, and the observed behavior on real images, demonstrate better generalization properties of the *U-Net*. To support this hypothesis, we perform an additional experiment. We add white Gaussian noise to the test images of our synthetic dataset and measure the performance of our pipeline using the different 2D pose estimation networks. In Fig. 5.18 we plot the MPJPE error under various levels of noise. Notably, the error of the *U-Net*-based pipeline increases slowly, while *Resnet 50*-based pipelines produce large errors already under small noise levels. This behavior supports our hypothesis that the *U-Net* architecture features better generalization properties.

5.4.3.2 Lifting Network: Parameter Ablation

In order to validate the architecture design choices of our *multi-branch* 3D pose lifting network, we perform an ablation study of two main parameters.

First, we find the optimal size of the embedding $\hat{\mathbf{z}}$, that encodes the 3D pose, the joint rotations, and the 2D pose uncertainty. Table 5.6 lists the MPJPE error using different sizes for $\hat{\mathbf{z}}$ for all three different heatmap estimation networks. Regardless of the choice of the heatmap estimation network, we find that $\hat{\mathbf{z}} \in \mathbb{R}^{50}$ produces the

best results. Smaller embeddings produce significantly higher errors, while larger embeddings only slightly impair the results.

Further, we study how the dimensions of the regressed heatmaps $\widetilde{\mathbf{HM}}$ influence the results, see 5.5. Unsurprisingly, we find that regressing the full heatmap produces the best results. This is in accordance with the experiments in Sec. 5.4, where we show that encoding uncertainty via regressing heatmaps helps over using them only as input.

To contribute towards fostering fairness in Computer Vision and Machine Learning we analyze the performance of the proposed models on our diverse dataset based

$\hat{\mathbf{z}}$ **size**	**Error (mm)**		
	ResNet50	ResNet50 (p)	UNet (p)
10	70.6	61.0	45.8
20	67.3	52.5	45.3
50	**54.7**	**51.1**	**41.0**
70	55.7	54.5	41.6
100	58.9	54.2	41.3
500	61.0	56.0	41.2

Table 5.5: Average reconstruction error per joint using Eq. 5.4, evaluated on the entire test-set when the model architecture differs based on the size of the embedding $\hat{\mathbf{z}}$. Increasing the latent space dimension produces worse results as with a larger dimensionality it starts to model noise together with the data.

HM size	**Error (mm)**		
	ResNet50	ResNet50 (p)	UNet (p)
48	**54.7**	**51.1**	**41.0**
36	57.8	59.6	44.2
24	59.9	57.7	43.8
16	61.2	56.8	41.4
8	61.4	56.7	41.7

Table 5.6: Average reconstruction error per joint using Eq. 5.4, evaluated on the entire test-set for different heatmap (HM) reconstruction sizes. Notice how little uncertainty information still has dramatic impact on the reconstruction accuracy.

on different skin tones. A comparison is shown in Table 5.7.

5.4.4 RESULTS ON EGOCENTRIC REAL DATASETS

In this section I am going to present a quantitative evaluation of our approach on egocentric real datasets — datasets with images captured by cameras — which accounts for a very limited set, including our own *x*R-EgoPose^R dataset. Our novel approach shows substantial improvements over all competing approaches.

5.4.4.1 *x*R-EgoPose^R

One of the main challenges when assessing an approach trained exclusively on synthetic datasets – like ours – is to find a way of evaluating how these results transform when the model is assessed on real data.

Generating real data in such condition is extremely hard since: *a)* it would require a mocap studio that can identify the 3D position of each joint to be used as ground truth; and *b)* that does not require the actor to wear a mocap suit, since this would alter the evaluation.

We have been able to generate a small real dataset withe two actors performing a subset of the actions contained in the main synthetic dataset for a total of 15000 frames, to quantitatively assess what is shown later only in the qualitative results. The reconstructing results for the different actions are shown in Table 5.8.

Skin tone	**Error (mm)**		
	ResNet50	ResNet50 (p)	UNet (p)
White	42.7	46.5	46.3
Light European	61.9	58.2	43.5
Dark European	63.6	52.0	35.6
Dark brown	22.5	28.7	27.5
Black	89.0	68.8	42.7

Table 5.7: Model evaluation based on skin tones.

Action name	**Average error (mm)**
Greeting	51.78
Talking	47.46
Playing Golf	68.74
Shooting	52.64
Upper Stretching	61.09
Throwing Arrow	88.54
Average	61.71

Table 5.8: Average reconstruction error per action on the test-set real data acquired from a mocap studio consisting of 2000 frames.

Approach	Evaluation error (mm)	Camera distance	Num Cameras
EgoCap *et al.* [122]	70	35 cm	2
Ours	58.2	2.1 cm	1

Table 5.9: Comparison of our proposed approach with a state-of-the-art egocentric pose estimator proposed by Rhodin *et al.* [122]. Since the dataset provided by [122] does not contain ground truth data compatible with the information required by our approach (monocular) and [122] requires a two-camera system dataset, we are not able to directly compare the two approaches on the same set of data. We therefore show a numerical evaluation of each approach on its own dataset.

5.4.4.2 Evaluation on EgoCap dataset

Rhodin *et al.* [122] presented one of the first egocentric 3D pose estimator from a multi-camera system. The objective comparison against this approaches is particularly challenging as model requires a multi-view dataset to be trained on, and the authors do not provide any 3D annotations for training data used to train their model on.

We show the results expressed in the paper in Table 5.9, however it is worth noticing that Rhodin *et al.* [122] has the advantage of using a stereo camera system where the cameras are placed 35 cm far form the face[1] (and thus better visibility conditions), while we count on a single monocular camera practically attached to the face.

[1] Camera distance is reported with respect to an average size noise and it is reported in centimeters

INDOOR	walking	sitting	crawling	crouching	boxing	dancing	stretching	waving	total (mm)
3DV'17 [162]	48.76	101.22	118.96	94.93	57.34	60.96	111.36	64.50	76.28
VCNet [163]	65.28	129.59	133.08	120.39	78.43	82.46	153.17	83.91	97. 85
Xu [157]	38.41	70.94	94.31	81.90	48.55	55.19	99.34	60.92	61.40
Ours - ResNet 50	**38.39**	61.59	69.53	51.14	**37.67**	**42.10**	58.32	**44.77**	48.16
Ours - U-Net (p)	45.83	**47.24**	**47.35**	**45.15**	48.72	47.00	**46.15**	46.45	**46.61**
OUTDOOR	walking	sitting	crawling	crouching	boxing	dancing	stretching	waving	total (mm)
3DV'17 [162]	68.67	114.87	113.23	118.55	95.29	72.99	114.48	72.41	94.46
VCNet [163]	84.43	167.87	138.39	154.54	108.36	85.01	160.57	96.22	113.75
Xu [157]	63.10	85.48	96.63	92.88	96.01	68.35	123.56	61.42	80.64
Ours - ResNet 50	**43.60**	85.91	83.06	69.23	69.32	**45.40**	76.68	**51.38**	60.19
Ours - U-Net (p)	53.96	**52.24**	**55.50**	**55.65**	**54.38**	54.48	**54.46**	56.12	**54.61**

Table 5.10: Quantitative evaluation on Mo^2Cap^2 dataset [157], for both indoor and outdoor test-sets. Our approach outperforms all competitors by more than **21.6%** (13.24 mm) on indoor data and more than **25.4%** (20.45 mm) on outdoor data when using only the provided synthetic data for training the model. Similarly to other experiments we provide in Sec 5.4.2, when using a pre-trained U-Net model with the configuration defined as in Sec 5.4.3.1, results improve even further: **24.9%** (14.79 mm) and **32.28%** (26.03 mm) respectively.

5.4.4.3 Evaluation on Mo2Cap2 dataset

This dataset consists of a similar size to our own *x*R-EgoPose, but of a lower quality in terms of both image resolution and realism of the data. To guarantee a fair comparison of the results, we have been provided by the authors the heatmaps generated by their own 2D estimator such that only the module taking care of lifting the points in 3D is assessed and no other factor is tampering the comparison, which was trained on *x*R-EgoPose's 3D data. Results of this comparison are shown for both indoor and outdoor scenarios in Table 5.10.

Our approach outperforms all competitors with a large margin (23.5% on average) on both indoor and outdoor test-sets, demonstrating indeed that the dataset introduced by Xu *et al.* [124] is not more challenging than our *x*R-EgoPose dataset, both in terms of actions or camera placement, which plays a big role in the amount of self-occlusion generated by the body.

An additional interesting experiments which can be introduced after proving all previous assumptions, is whether a pipeline approach can indeed exploit the benefit

which consists in increasing the dataset size with partially labelled data, leading to result improvements.

Such statement is often claimed as one of the best benefits of pipeline approaches. E.g. new heatmaps without having the corresponding 3D annotations. To prove this, we add heatmaps belonging to the indoor set when training the model on the mo2cap2 dataset and perform the identical evaluation as previously done on the outdoor test-set. The final error after performing this data augmentation reduces from *60.19 mm to 60.05 mm* proving the desired effect.

5.4.5 EVALUATION ON FRONT-FACING-CAMERA DATASETS

The proposed architecture proves to be working particularly well when the 2D pose represented in the image is affected by a large amount of self-occlusion, and the reconstructed 3D pose is able to handle that without being heavily compromised. However, there is a very related – but at the same time different – problem which consists in solving the task of 3D pose detection from a front facing camera.

As already mentioned, although the final goal of both tasks is identical, the conditions under which the algorithm has to work are extremely different, since front facing camera approaches:

- the "pixel density" of each joint represented in the image is the same
- very often they assume a weak-perspective camera model, since this doesn't seem to affect significantly the results
- the amount of self occlusion is very limited compared to egocentric camera-views.

One significant effect of all these differences is the possibility to download an already available dataset or even a pre-train model to be used as an initialization step. Regardless of these conditions, we expect our architecture to maintain good performance on this task as well.

Protocol #1	Chen [87]	Hossain [165]*	Dabral [166]*	Tome [16]	Moreno [83]	Kanazawa [91]	Zhou [167]	Jahangiri [168]	Mehta [162]	Martinez [82]	Fang [169]	Sun [72]	Sun [152]	**Ours**
Errors (mm)	114.2	51.9	52.1	88.4	87.3	88.0	79.9	77.6	72.9	62.9	60.4	59.1	**49.6**	51.3
Protocol #2	Yasin [170]	Hossain [165]*	Dabral [166]*	Rogez [139]	Chen [87]	Moreno [83]	Tome [16]	Zhou [167]	Martinez [82]	Kanazawa [91]	Sun [72]	Fang [169]	Sun [152]	**Ours**
Errors (mm)	108.3	42.0	36.3	88.1	82.7	76.5	70.7	55.3	47.7	58.8	48.3	45.7	**40.6**	42.3

Table 5.11: Comparison with other state-of-the-art approaches on the Human3.6M dataset (front-facing cameras). Approaches with * make use of temporal information. No specific modifications have been applied to our architecture: UNet 2D pose detector pre-trained on ImageNet has been used to estimate joint-heatmaps fed through our dual-branch auto-encoder architecture, since rotation information is not available for these data.

For this evaluation, we chose the Human3.6M dataset [129, 164]. We used two evaluation protocols. *Protocol 1* has five subjects (S1, S5, S6, S7, S8) used in training, with subjects (S9, S11) used for evaluation. The MPJPE error is computed on every 64th frame. *Protocol 2* contains six subjects (S1, S5, S6, S7, S8, S9) used for training, and the evaluation is performed on every 64th frame of Subject 11 (Procrustes aligned MPJPE is used for evaluation). The results are shown in Table 5.11 from where it can be seen that our approach is on par with state-of-the-art methods, scoring second overall within the non-temporal methods.

5.4.6 DATA-AUGMENTATION

An important advantage of our architecture is that the model can be trained on a mix of 3D and 2D datasets simultaneously: if an image sample only has 2D annotations but no 3D ground truth labels, the sample can still be used, only the heatmaps will contribute to the loss. We evaluated the effect of adding additional images with 2D but no 3D labels on both scenarios: egocentric and front-facing cameras. In the egocentric case we created two subsets of the *x*R-EgoPose test-set. The first subset contained 50% of all the available image samples with both 3D and 2D labels. The second contained 100% of the image samples with 2D labels, but only 50% of the 3D labels. Effectively the second subset contained twice the number of images with 2D annotations only. Table 5.12 compares the results between the subsets, where it can be seen that the final 3D pose estimate benefits from additional 2D

annotations. Equivalent behavior is seen on the Human3.6M dataset. Table 5.13 shows the improvements in reconstruction error when additional 2D annotations from COCO [171] and MPII [161] are used.

5.4.7 QUALITATIVE RESULTS

In this section, a qualitative evaluation of the approach is provided by analyzing the reconstructed poses on different use cases and datasets.

5.4.7.1 Encoding uncertainty in the latent space

One of the main statements made about the proposed model is about the ability of generating a better and more robust latent space that is able not only to express the position of each individual joint composing a pose, but also its uncertainty of estimation. This is enforced by adding the second branch that reconstructs the set of 2D heatmaps (one per joint) from the latent vector. It has been proved numerically that this novel architecture generates much better results compared to its single-branch version (see Sec. 5.4.2), however something that needs to be proven is whether or not the reconstructed uncertainties correspond to the original ones.

To this end, Figure 5.19 shows how the heatmaps decoded from the latent space closely resemble the ground truth ones with per-estimation variations of the uncertainties, especially when considering the same joint. This therefore demonstrates

3D	2D	Error (mm)
50%	50%	68.04
50%	100%	63.98

Table 5.12: Availability of training data for *x*E-EgoPose dataset

Training dataset	Error (mm)
H36M	67.9
H36M + COCO + MPII	53.4

Table 5.13: Availability of training data for front facing camera datasets

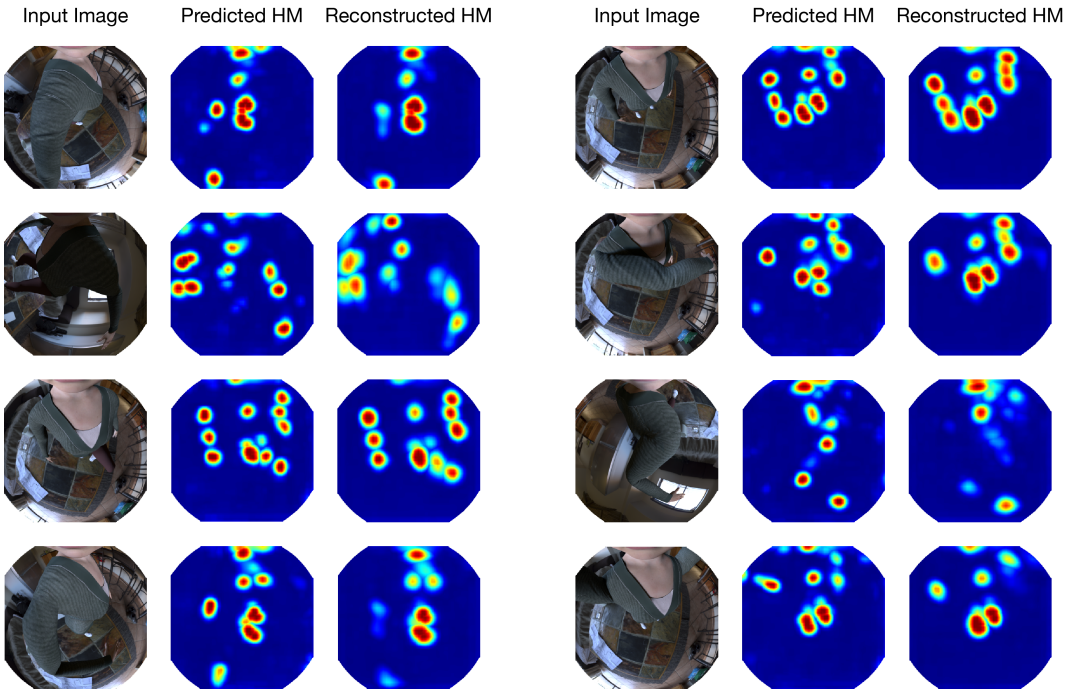


Figure 5.19: Reconstruction of Heatmaps through the multi-branch AE that encodes the heatmaps into a latent vector z, forcing the latent space to learn both to regress the joint position and also to account for the joint uncertainty. Here we are making sure that the model is not learning a constant uncertainty per joint (e.g. uncertainty of wrist predictions is always worse than the one for the shoulders) but rather the model is properly learning how to encode uncertainty based on the input pose.

the ability of our approach to encode the uncertainty of the 2D heatmap predictions in the latent vector, to be used for better 3D pose estimations.

5.4.7.2 Character animation

The proposed architecture has, among some of the benefits already introduced, the ability of generating different representations for the pose: i.e. *a)* 3D joint positions and *b)* local joint rotations with respect to the parent node.

In the experiment section 5.4.2 we have so far addressed the evaluation of the precision for the 3D joint predictions. In Figure 5.20 we instead focus on the rotation representation of the pose, showcasing few rendered reconstructions of a larger sequence to highlight how this model can be used to directly drive an avatar, when the predicted local joint rotations are used to generate the animation.

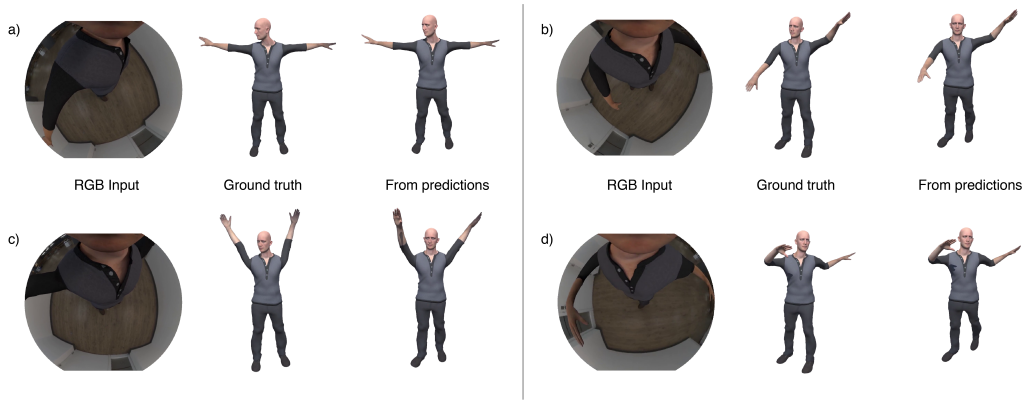


Figure 5.20: Character animation from the joint local rotation predictions computed from the input image. Note how the model is able to retrieve most of the desired information even when limbs fall outside the camera field of view.

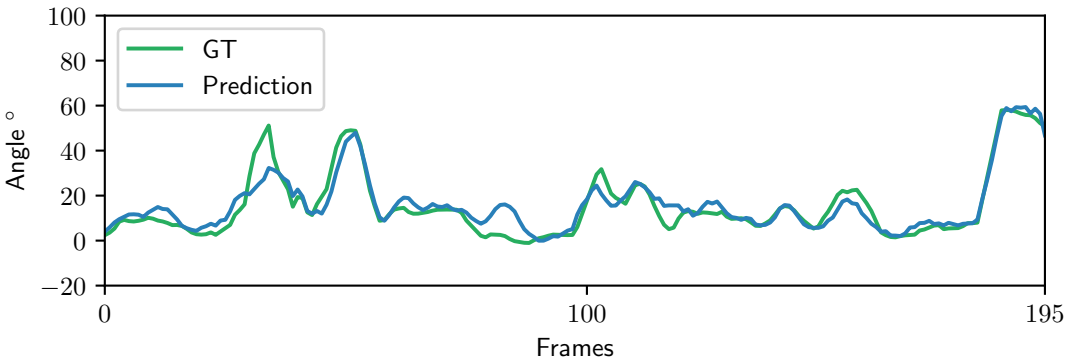


Figure 5.21: Analysis of the angle predictions through time for the Right Foot in sequence of the test-set.

To better visualize the stability of the local rotations through time, Figure 5.21 shows the variation of the right-foot angle — a joint that moves with higher frequency than other joints in the sequence — through time, compared to the ground truth local rotation.

5.4.7.3 Qualitative results on our datasets

A qualitative evaluation of the performance of the proposed architecture is shown in Figure 5.22 on synthetic data from the *xR-EgoPose dataset* and on real captured with the Headset-Mounted Camera. As shown in the figure, the model can handle a large amount of self occlusions and is able to perform good reconstructions even

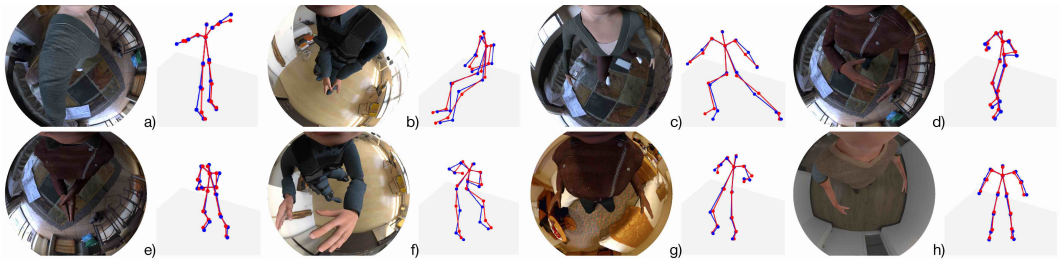


Figure 5.22: Qualitative results on synthetic images from our *x*R-EgoPose dataset. Blue are ground truth poses and red predictions;

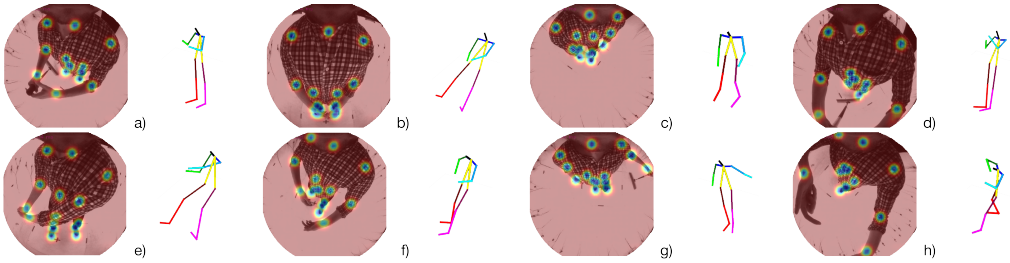


Figure 5.23: Qualitative results on real images from our *x*R-EgoPose^R dataset.

with very difficult poses. Note that due to the diversity of the generated dataset, it is also possible to reliably identify the pose with actors wearing clothes matching the background.

Additional results for our *x*R-EgoPose^R dataset are shown in Figure 5.23 where both 2D heatmap predictions and 3D reconstructions are displayed.

5.4.7.4 Reconstructions on Mo2Cap2

A qualitative evaluation of the performance of the proposed architecture is shown in Figure 5.24 on **real data** capture outside, from the *Mo2Cap2 dataset*. As shown in the figure, the model is able to retrieve correctly the pose from real data even with very difficult poses. Since the authors of the Mo2Cap2 dataset don't provide camera calibration parameters, for visualization reasons we plot all the poses according to the same reference system (not the camera one).

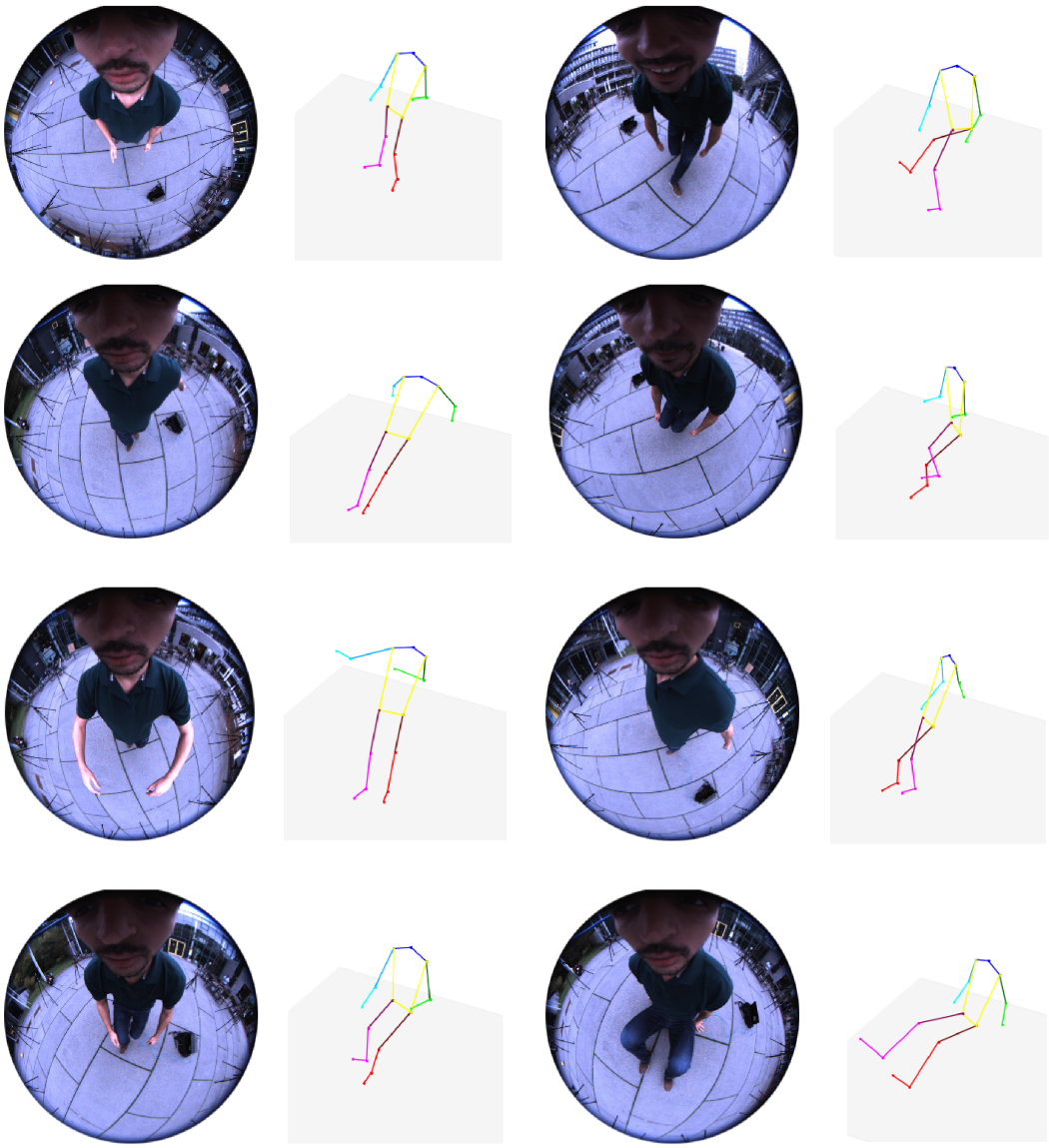


Figure 5.24: Qualitative results on **real** images captured outside from the Mo2Cap2 dataset.

5.4.7.5 Latent Space

In Figure 5.4 we presented the projected latent space (using t-SNE) of the model trained both with a standard AE and with the proposed dual-branch AE. Carefully analyzing figure *b*, some "string-like" shapes appear which need to be investigated. The analysis of some of those poses is presented in Figure 5.25 where we show how the reconstructions are consistent and those latent vectors are not the result of any artifact, but appear only due to some short sequences in the dataset, where one

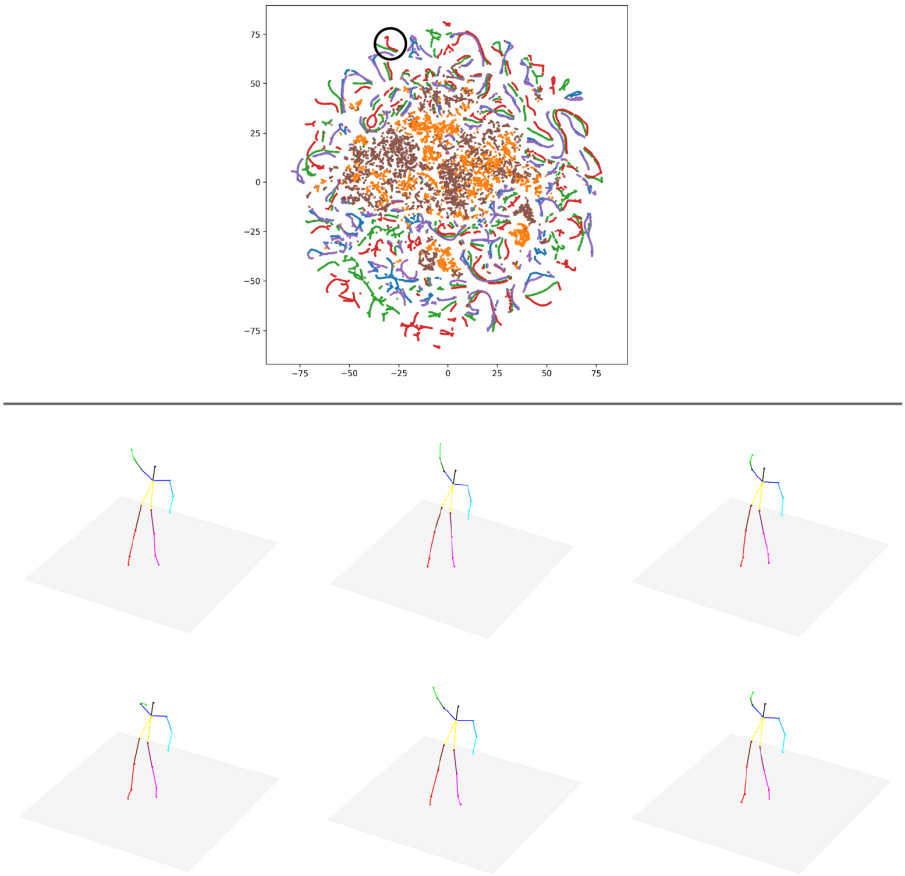


Figure 5.25: Analysis of the latent space generated with the proposed dual-branch AE, by inspecting a small area (highlighted in black), specifically one character (red), to see the reconstructed poses. Among those poses, we sample few representative ones to show. This is done to investigate the string-like shape of a set of poses. The poses are close together since they share most of the joint positions, and only few joints move (as expected).

character is performing some repetitive movements with most of the joints in the same position for the entire duration of the small sequence.

5.5 CONCLUSION

We have presented a solution to the problem of 3D body pose estimation from a monocular camera installed on a HMD. Our solution involves a fully differentiable network that estimates input images to heatmaps, and from heatmaps to 3D pose via an dual-branch autoencoder. This particular autoencoder architecture was fundamental for training and generalization purposes. We have also introduced the EgoHMD-Dataset, a new large scale photorealistic synthetic dataset that was essential for training and will be made publicly available to promote research in this exciting area. While our results are state-of-the-art, there are a few failures cases due to extreme occlusion and the inability of the system to measure hands when they are out of the field of view. This could be potentially be remedied by adding additional cameras to the headset to cover currently unseen areas. Further improvements in accuracy could be potentially be achieved by the use of a stereo system. These two improvements are the focus of our future work.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

To conclude, we summarise the main contributions of this thesis by highlighting the strengths and weaknesses of the different proposed solutions, as well as potential directions for future research, taking into account the current developments within the filed of study.

We have tried to keep the related work in section 2 up-to-date — including new approaches published after the publication of our contributions — in this very dynamic and fast-moving research area.

6.1 POSE FROM MONOCULAR IMAGE

The first core chapter (Chapter 3) of this thesis introduces a novel approach for 3D human pose estimation from a single input image, with a novel hybrid neural network solution.

Prior research focused either on purely end-to-end or pipeline approaches, each with their own limitations. The former has generally better performance but suffers data collection limitations and therefore generalization on new data, since it requires $\{x, y\}$ data samples with pairs of input images with their corresponding 3D annotated labels. The latter, on the other hand, is more flexible since it allows to use data sources from multiple datasets — captured in different domains — at

a cost of worse performance, due to not exploiting the interdependence between 2D and 3D data. We therefore bridged the gap between these two approaches by introducing a *hybrid* architecture which has the flexibility of pipeline approaches with better performance of end-to-end methods due to the exploitation of 2D-3D inter-dependency.

Our quantitative evaluation (Sec 3.4) demonstrates how this novel architecture outperformed on average all competing approaches, as well as dominating almost all action types on the challenging Human3.6M dataset [129], even when comparing against approaches exploiting temporal information. Most significantly, we found that our proposed solution performs well on images-in-the-wild (as shown in the Sec. 3.4.3) which is possible due to the design of the proposed architecture which can easily be trained on a collection of complementary datasets with partial annotations, which could have not been used in end-to-end approaches.

Our novel architecture shows the importance of thinking in 3D even for 2D pose estimation within a single image, with the iterative 3D model demonstrating better 2D accuracy than Convolutional Pose Machines [15], the iterative 2D approach it is based upon.

After the publication of our approach many new methods have followed this idea of designing model to better exploit data from different domains. So far 2.5D models (e.g. the approach by Pavlakos [70]) have demonstrated the same flexibility of a hybrid architecture with more end-to-end like performance, with results that can only be incrementally be improved at this point if focusing on bridging the gap between 2D and 3D.

An area of improvement that has been largely underrated instead is bridging the gap between skeleton representations. There are several 3D datasets currently available to the research community. Unfortunately, each has their own skeletal representation (number of joints and their connectivity) which makes it hard to fuse all those

data in a unified dataset. The future work for this approach would be to create a skeleton-independent representation to learn from, which will produce a model than can finally unify 2D and 3D from different domains as well as different representations.

6.2 POSE FROM MULTI-CAMERA VIEWS

Our work on a hybrid multi-view camera system introduced in Chapter 4 once again pushed the state-of-the-art results in 3D pose estimation from a multi-view set-up by exploiting dependencies between 2D and 3D data.

Here we demonstrated how the design decision introduced in the previous chapter for the monocular architecture can be extended for a multi-view set with similar performance improvement. Furthermore, given the noticeable improvement in accuracy obtained by using multiple cameras rather than just one, we have then assessed the ability of the proposed multi-view3D human pose estimator to meaningfully label unlabeled data that can be used at train time in order to achieve better performance.
We have shown how, with this novel architecture, using both data with labels (supervised training) and data without any associated annotations (unsupervised) leads to the best performance.

Similarly to what has been seen on monocular 3D pose estimation approaches, a major result improvement could be seen if a skeleton representation independent approach would be introduced. However, unlike the monocular case, not much effort has been put by the community in designing better multi-view systems.

An interesting extension of this approach would be to have a definition where cameras are allowed to move and zoom in on a person. Nowadays, mocap-studios are being used by VFX studios to capture actors performing various actions. Rather than relying on a static system, we could think of a system that adapts based on what

the actor is doing, to ensure better reconstruction accuracy, by focusing the camera on the actor instead of "wasting" pixels on background, non-interesting ares, etc.

6.3 POSE FROM EGOCENTRIC PERSPECTIVE

The interest of the community and the potential applications of VR/AR has led us in designing a 3D pose estimator from egocentric perspectives. Here, the already challenging 3D pose estimation problem has been pushed even further, with a far more challenging configuration where occlusions and distortions play a very important role.

Unlike the monocular and multi-view scenarios introduced in the previous chapters, where large amount of data is already available and "just" needs to be properly and carefully used, in this specific scenario this is not the case. Therefore, we had to rely on highly realistic synthetic data generation which would allow us to still generalize on in-the-wild images.

Additionally, due to the consequences of an embedded camera on the headset-mounded-device (i.e. distortion and occlusion), we had to design a novel neural network architecture able to deal with such harsh conditions. We introduced a multi-branch AE architecture capable of large levels of occlusions and self-occlusion combined with fish-eye distortions cameras.

We have proven how this novel architecture is significantly better than competing approaches, drastically outperforming them even on their own dataset and how this approach is capable of generalizing on images-in-the-wild under a variety of different conditions. Furthermore, we have extensively analyzed the performance of such approach, trying to understand the current limitations both from a model perspective as well as a hardware one.
From our analysis we have been able to assess the limitation of our current model: visibility of the limbs is a problem. For a fully product-ready method, the hands

need to be visible even when the user if performing weird motions, which is not the case in the current configuration. Future work therefore should be focusing on a multi-view system for which the hands are more visible and more accuracy on the hand reconstructions can be guaranteed, especially important due to the current main nature of these devices (i.e. gaming).

6.4 SUMMARY

In summary, this thesis is a collection of four research contributions, aiming at using all possible information that can be extracted from the data to ensure better performance for 3d human pose estimation on a variety of working conditions.

We have unequivocally demonstrated how having more data, captured from different domains, benefits the performance of the model and we have shown some techniques to due so. Using 3d labels, when available, together with 2D annotations improve not only 3D prediction accuracy, but 2D reconstructions we well.

Finally, we have proven how the uncertainty of the estimation is incredibly valuable in extreme working conditions for the model to operate in, e.g. with egocentric human pose estimation, where the uncertainty plays a fundamental role in achieving accurate reconstructions.

Appendices

APPENDIX A

3D LIFTER — GRADIENT PROPAGATION

A.1 COMPUTING DERIVATIVES

As discussed by the Convolution Pose Machine paper [15], recurrent-like architectures such as ours have problems with *vanishing gradients* and for effective training they require an additional loss function to be defined for each layer, that independently drives each individual layer to return correct predictions regardless of how this information is used in subsequent layers.

Before we give the derivation of the gradients it should be emphasized that it is entirely possible to train the network without using them – in fact similar results can be obtained by only using the 3D lifting for the forward pass, and not back-propagating the lifting derivatives through the rest of the network.

As the additional layers make use of custom Python-based derivatives rather than an efficient implementation, for computational reasons it might preferable to avoid this step. Nonetheless for completeness we include the derivatives.

There are two reasons the gradients are unneeded: *a)* the lifting 3D model we use makes its best predictions when the 2D predictions of the same layer are closest to ground truth, and this is a constraint naturally enforced by the objective of equation 3.9 of the main paper. Further, as with Convolutional Pose Machines [15] our architecture suffers from problems with vanishing gradients. To overcome this Wei *et al.* [15] defined an objective at each layer, which acted to locally strengthen the

gradients. However, a side effect of this multi-stage objective is that most of the effects of back-propagation happen locally and gradients back-propagated from other layers have little effect on the learning. This makes subtle interactions between layers less influential, and forces the learning process to concentrate on simply making accurate 2D predictions in each layer.

We first give the results for computing the gradients of sparse predicted locations $\hat{Y}$ from Y (see section 5 of the main paper), before discussing the gradients induced on the confidence maps by these sparse locations.

A.1.1 LANDMARK GRADIENTS

In the interests of readability we neglect the use of indices to indicate stages, the reader should assume that all variables are taken from the same stage. Similarly, when dealing with a mixture of Gaussians, as we are only interested in computing a sub-gradient, the reader should assume that the best model has already been selected in the forward pass and we are computing gradients using only this model.

Recall (section 5 of main body of paper) that the mapping from the initial landmarks Y to the projected 3D proposals $\hat{Y}$ is given by

$$\hat{Y} = \Pi\mathbf{R}(\mu + a \cdot \mathbf{e}) \tag{A.1}$$

where

$$\mathbf{R}, a = \underset{\{\mathbf{R}^* \in \mathscr{R}, a^* \in \mathbb{R}^J\}}{\arg\min} ||Y - \Pi\mathbf{R}^*(\mu + a^* \cdot \mathbf{e})||_2^2 + (\sigma \cdot a^*)^2 \tag{A.2}$$

where $\mathscr{R}$ is a discrete set of rotations we exhaustively minimize over, and J is the number of bases in $\mathbf{e}$. Owing to the use of discrete rotations, this mapping from Y to $\hat{Y}$ is a piece-wise smooth approximation of the smooth function defined over a continuous $\mathscr{R}$, and sub-gradients can be induced by fixing R to its current state. Hence:

$$\frac{d\hat{Y}}{da} = \Pi \mathbf{Re} \tag{A.3}$$

For the remainder of the section, and to compact notation we will write E for the matrix of size $2L \times J$ (L the number of landmark points and J being the number of bases in $\mathbf{e}$) formed by unwrapping tensor $\Pi\mathbf{Re}$. Similarly, we will unwrap the $2 \times L$ matrices Y and $\hat{Y}$ and write them as y and $\hat{y}$. We also write p for the vector representing the unwrapped set of 2D landmark positions $\Pi\mathbf{R}\mu$.

We will use $[y,0]$ for the vector formed by vector y followed by J zeros, and $\bar{E}$ for the matrix of size $(2L+J) \times J$ formed by concatenating E with the matrix that has values σ along the diagonals and zero everywhere else. We can rewrite equation (A.3) in its new notation as:

$$\frac{d\hat{y}}{da} = E \tag{A.4}$$

and given R, we can rewrite equation (A.2) as

$$a = \underset{a* \in \mathbb{R}^J}{\arg\min} \, |[y,0] - [p,0] - a^* \bar{E}||_2^2 \tag{A.5}$$

or

$$a = [(y-p),0]\bar{E}^\dagger \tag{A.6}$$

with $\bar{E}^\dagger$ continuing to represent the pseudo-inverse of $\bar{E}$. Hence

$$\frac{da}{d[y,0]} = \bar{E}^\dagger \tag{A.7}$$

and

$$\frac{d\hat{y}}{dy} = \frac{d\hat{y}}{da}\frac{da}{dy} = E\,\bar{E}^t \tag{A.8}$$

where $\bar{E}^t$ is the truncation of $\bar{E}^\dagger$.

A.1.2 MAPPING HM GRADIENTS

The coordinates of each predicted landmark $\hat{Y}_p$ induce a Gaussian in the belief map $\hat{b}_p$. So a change in the x component of $\hat{Y}_p$ induces an update which is equivalent to a difference of Gaussians.

$$\frac{d\hat{b}_p}{d\hat{Y}_p^x} \approx \frac{G(\hat{Y}_p{}^{(x)} + \delta_x) - G(\hat{Y}_p{}^{(x)} - \delta_x)}{2\delta_x} \tag{A.9}$$

and the same for the y component as well. For computational purposes we take δ_x as one pixel. As such, an induced gradient on the projected belief map near the predicted location $\hat{Y}$ $\hat{b}_p$ induces an updating of $\hat{Y}$ that is propagated through to Y using the sub-gradients described in equation (A.8).

A.1.2.1 Updating B

Writing B for the the set of all b_p, and assuming Y_p is not in the right location, i.e. given updates $\Delta\hat{B}$ on $\hat{B}$ such that

$$\Delta\hat{B}.\frac{d\hat{B}}{d\hat{Y}}\frac{d\hat{Y}}{dY_p} \neq 0,$$

any update of b in which we decrease the belief at $b^p_{Y_p}$ and increase anywhere else is a valid sub-gradient. We choose as a sensible update a negative step at b_p of magnitude $m = k||\Delta\hat{B}.\frac{d\hat{B}}{d\hat{Y}}\frac{d\hat{Y}}{dY_p}||$ and a positive update for each element Y of of B_p of the magnitude $m \cdot N(Y, \sigma^2)$ in the quadrant of a Gaussian of the same width used to generate $\hat{b}$ (i.e. $\sigma = 1$ see section 5.6 of main paper) and with the same direction as $\Delta\hat{B}.\frac{d\hat{B}}{d\hat{Y}}\frac{d\hat{Y}}{dY_p}$ in each x and y coordinate.

BIBLIOGRAPHY

[1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[2] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6050–6059, 2017.

[3] *Body Pose Annotations Correction*, 2016.

[4] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, 100(1):67–92, 1973.

[5] Tom M Mitchell. Machine learning, 1997.

[6] 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*, pages 561–578. Springer, 2016.

[7] Denis Tome, Matteo Toso, Lourdes Agapito, and Chris Russell. Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture. In *2018 international conference on 3D vision (3DV)*, pages 474–483. IEEE, 2018.

[8] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: What early learning tells us about the mind*. William Morrow Paperbacks, 2000.

[9] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[10] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

[11] RICHARD Bellman. Dynamic programming, princeton univ. *Prese Princeton, 19S7*, 1957.

[12] F Rosenblatt. The perceptron–a perceiving and recognizing automaton, cornell aeronautical laboratory. Technical report, Report 85-460-1, 1957.

[13] Jose-Manuel Alonso and Yao Chen. Receptive field. *Scholarpedia*, 4(1):5393, 2009.

[14] Dang Ha The Hien. A guide to receptive field arithmetic for convolutional neural networks. `https://medium.com/mlreview/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-` 2017.

[15] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *arXiv preprint arXiv:1602.00134*, 2016.

[16] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CVPR 2017 Proceedings*, pages 2500–2509, 2017.

[17] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7728–7738, 2019.

[18] D. Tome, T. Alldieck, P. Peluse, G. Pons-Moll, L. Agapito, H. Badino, and F. De la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[19] G Hinton. Using relaxation to find a puppet. In *Proceedings of the 2nd Summer Conference on Artificial Intelligence and Simulation of Behaviour*, pages 148–157. IOS Press, 1976.

[20] Alex Pentland and Bradley Horowitz. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):730–742, 1991.

[21] Dariu M Gavrila. Vision-based 3-d tracking of humans in action. 1996.

[22] Dirk Ormoneit, Hedvig Sidenbladh, Michael J Black, and Trevor Hastie. Learning and tracking cyclic human motion. In *Advances in Neural Information Processing Systems*, pages 894–900, 2001.

[23] Raquel Urtasun, David J Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 238–245. IEEE, 2006.

[24] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2007.

[25] 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.

[26] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[27] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[28] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, volume 2, page 5, 2010.

[29] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013.

[30] 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, 2013.

[31] 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.

[32] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[33] Matthias Dantone, Juergen Gall, Christian Leistner, and Luc Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.

[34] Leonid Karlinsky and Shimon Ullman. Using linking features in learning non-parametric part models. In *European Conference on Computer Vision*, pages 326–339. Springer, 2012.

[35] Xiangyang Lan and Daniel P Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 470–477. IEEE, 2005.

[36] Leonid Sigal and Michael J Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2041–2048. IEEE, 2006.

[37] Yang Wang and Greg Mori. Multiple tree models for occlusion and spatial constraints in human pose estimation. In *European Conference on Computer Vision*, pages 710–724. Springer, 2008.

[38] Min Sun and Silvio Savarese. Articulated part-based model for joint object detection and pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 723–730. IEEE, 2011.

[39] Yuandong Tian, C Lawrence Zitnick, and Srinivasa G Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *European Conference on Computer Vision*, pages 256–269. Springer, 2012.

[40] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[41] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[42] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016.

[43] Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *European Conference on Computer Vision*, pages 33–47. Springer, 2014.

[44] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[45] Hsi-Jian Lee and Zen Chen. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168, 1985.

[46] Camillo J Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 677–684. IEEE, 2000.

[47] Vasu Parameswaran and Rama Chellappa. View independent human body pose estimation from a single perspective image. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–16. IEEE, 2004.

[48] Carlos Barrón and Ioannis A Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81(3):269–284, 2001.

[49] Xiaochuan Fan, Kang Zheng, Youjie Zhou, and Song Wang. Pose locality constrained representation for 3d human pose reconstruction. In *European Conference on Computer Vision*, pages 174–188. Springer, 2014.

[50] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision*, pages 573–586. Springer, 2012.

[51] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1455, 2015.

[52] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000.

[53] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[54] Paulo Gotardo and Aleix Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[55] M. Lee, J. Cho, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[56] Jungchan Cho, Minsik Lee, and Songhwai Oh. Complex non-rigid 3d shape recovery using a procrustean normal distribution mixture model. *International Journal of Computer Vision*, 117(3):226–246, 2016.

[57] C. Wang, Y. Wang, Z. Lin, , A. Yuille, and W. Gao. Robust estimation of human poses from a single image. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[58] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. Sparse representation for 3d shape estimation: A convex relaxation approach. *arXiv preprint arXiv:1509.04309*, 2015.

[59] Ruiqi Zhao, Yan Wang, and Aleix Martinez. A simple, fast and highly-accurate algorithm to recover 3d shape from 2d landmarks on a single image. *arXiv preprint arXiv:1609.09058*, 2016.

[60] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2006.

[61] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, 2004.

[62] Carl Henrik Ek, Philip H. S. Torr, and Neil D. Lawrence. Gaussian process latent variable models for human pose estimation. In Andrei Popescu-Belis, Steve Renals, and Hervé Bourlard, editors, *MLMI*, volume 4892 of *Lecture Notes in Computer Science*, pages 132–143. Springer, 2007.

[63] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 2006.

[64] L. Sigal, R. Memisevic, and D. Fleet. Shared kernel information embedding for discriminative inference. In *CVPR*, 2009.

[65] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[66] Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of*

the IEEE International Conference on Computer Vision, pages 2848–2856, 2015.

[67] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. *arXiv preprint arXiv:1609.05317*, 2016.

[68] 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)*, 2016.

[69] Bugra Tekin, Pablo Marquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[70] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1263–1272. IEEE, 2017.

[71] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[72] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017.

[73] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision*, 2017.

[74] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[75] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2016.

[76] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in Neural Information Processing Systems*, pages 1736–1744, 2014.

[77] Arjun Jain, Jonathan Tompson, Mykhaylo Andriluka, Graham W Taylor, and Christoph Bregler. Learning human pose estimation features with convolutional networks. *arXiv preprint arXiv:1312.7302*, 2013.

[78] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1913–1921, 2015.

[79] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, and Kostas Daniilidis. Sparse representation for 3d shape estimation: A convex relaxation approach. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1648–1661, 2017.

[80] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016.

[81] Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Bayesian image based 3d pose estimation. In *European Conference on Computer Vision*, pages 566–582. Springer, 2016.

[82] 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*, volume 206, page 3, 2017.

[83] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1561–1570. IEEE, 2017.

[84] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3d pose sequence machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 810–819, 2017.

[85] Edgar Simo-Serra, Arnau Ramisa, Guillem Alenyà, Carme Torras, and Francesc Moreno-Noguer. Single image 3d human pose estimation from noisy observations. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2673–2680. IEEE, 2012.

[86] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. *arXiv preprint arXiv:1511.09439*, 2015.

[87] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[88] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[89] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. Synthetic occlusion augmentation with volumetric heatmaps for the 2018 eccv posetrack challenge on 3d human pose estimation. *arXiv preprint arXiv:1809.04987*, 2018.

[90] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[91] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.

[92] 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*, pages 7307–7316, 2018.

[93] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016.

[94] Hsiao-Yu Fish Tung, Adam Harley, William Seto, and Katerina Fragkiadaki. Adversarial inversion: Inverse graphics with adversarial priors. *arXiv preprint arXiv:1705.11166*, 2017.

[95] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5242–5252, 2017.

[96] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[97] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1477–1485. IEEE, 2019.

[98] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1146–1161, 2019.

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

[100] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10133–10142, 2019.

[101] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018.

[102] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3d sensing of multiple people in natural images. In *Advances in Neural Information Processing Systems*, pages 8410–8419, 2018.

[103] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[104] Yebin Liu, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *CVPR 2011*, pages 1249–1256. IEEE, 2011.

[105] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International journal of computer vision*, 87(1-2):75, 2010.

[106] Magnus Burenius, Josephine Sullivan, and Stefan Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3618–3625, 2013.

[107] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3d pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014.

[108] Ahmed Elhayek, Edilson de Aguiar, Arjun Jain, Jonathan Tompson, Leonid Pishchulin, Mykhaylo Andriluka, Christoph Bregler, Bernt Schiele, and Christian Theobalt. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3810–3818. IEEE, 2015.

[109] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017.

[110] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6988–6997, 2017.

[111] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7718–7727, 2019.

[112] Abdolrahim Kadkhodamohammadi and Nicolas Padoy. A generalizable approach for multi-view 3d human pose regression. *arXiv preprint arXiv:1804.10462*, 2018.

[113] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4342–4351, 2019.

[114] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6040–6049, 2020.

[115] Alireza Fathi, Ali Farhadi, and James M. Rehg. Understanding egocentric activities. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[116] Minghuang Ma, Haoqi Fan, and Kris M. Kitani. Going deeper into first-person activity recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1894–1903, 2016.

[117] Congqi Cao, Yifan Zhang, Yi Wu, Hanqing Lu, and Jian Cheng. Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[118] Grégory Rogez, James S Supancic, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4325–4333, 2015.

[119] H. Yonemoto, K. Murasaki, T. Osawa, K. Sudo, J. Shimamura, and Y. Taniguchi. Egocentric articulated pose tracking for action recognition. In *International Conference on Machine Vision Applications (MVA)*, 2015.

[120] Hao Jiang and Kristen Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3509. IEEE, 2017.

[121] Maria Amer, Saeid Vosoughi Amer, and A Maria. Deep 3d human pose estimation under partial body presence. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018.

[122] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):162, 2016.

[123] Young-Woon Cha, True Price, Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, Zihe Qin, Hyounghun Kim, Zhaoqi Su, Yebin Liu, et al. Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. *IEEE transactions on visualization and computer graphics*, 2018.

[124] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo2cap2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *arXiv preprint arXiv:1803.05959*, 2018.

[125] Timo von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, volume 36, pages 349–360. Wiley Online Library, 2017.

[126] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K Hodgins. Motion capture from body-mounted cameras. In *ACM Transactions on Graphics (TOG)*, volume 30, page 31. ACM, 2011.

[127] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. Human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[128] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12.

[129] 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 transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[130] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[131] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[132] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[133] Nikolaos Pitelis, Chris Russell, and Lourdes Agapito. Learning a manifold as an atlas. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1642–1649. IEEE, 2013.

[134] Chunyu Wang, John Flynn, Yizhou Wang, and Alan L Yuille. Representing data by a mixture of activated simplices. *arXiv preprint arXiv:1412.4102*, 2014.

[135] A Vedaldi, Y Jia, E Shelhamer, J Donahue, S Karayev, J Long, and T Darrell. Convolutional architecture for fast feature embedding. *Cornell University, arXiv: 1408.5093 v12014*, 2014.

[136] Bugra Tekin, Xiaolu Sun, Xinchao Wang, Vincent Lepetit, and Pascal Fua. Predicting people's 3d poses from short sequences. *arXiv preprint arXiv:1504.08200*, 2015.

[137] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Fusing 2d uncertainty and 3d cues for monocular body pose estimation. *arXiv preprint arXiv:1611.05708*, 2016.

[138] H. Yasin, U. Iqbal, B. Kruger, 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.

[139] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in Neural Information Processing Systems*, pages 3108–3116, 2016.

[140] Chris Russell, Rui Yu, and Lourdes Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *European conference on computer vision*, pages 583–598. Springer, 2014.

[141] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1272–1279. IEEE, 2013.

[142] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[143] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[144] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. *arXiv preprint arXiv:1705.04098*, 2017.

[145] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[146] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[147] Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Learning latent representations of 3d human pose with deep neural networks. *International Journal of Computer Vision*, pages 1–16, 2018.

[148] DeepMotion. How to make 3 point tracked full-body avatars in vr, https://medium.com/@deepmotioninc/how-to-make-3-point-tracked-full-body-avatars-in-vr-34b3f6709782, last accessed on 2018-11-16.

[149] U. Hess, K. Kafetsios, H. Mauersberger, C. Blaison, and CL. Kessler. Signal and noise in the perception of facial emotion expressions: From labs to life. *Pers Soc Psychol Bull*, 42(8), 2016.

[150] James T Reason and Joseph John Brand. *Motion sickness.* Academic press, 1975.

[151] Grégory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images. *CoRR*, abs/1803.00455v1, 2018.

[152] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[153] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[154] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[155] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image.

In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[156] Carnegie mellon university motion capture database.

[157] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo^2Cap^2 : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.

[158] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[159] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[160] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4918–4927, 2019.

[161] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[162] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV)*, 2017.

[163] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Chris-

tian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.

[164] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[165] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision*, pages 69–86. Springer, 2018.

[166] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, and Arjun Jain. Structure-aware and temporally coherent 3d human pose estimation. *arXiv preprint arXiv:1711.09250*, 2017.

[167] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[168] Ehsan Jahangiri and Alan L Yuille. Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 805–814, 2017.

[169] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[170] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[171] 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*, pages 740–755. Springer, 2014.