Thèse n° 8429

Efficient Depth-based Deep Learning Methods for Multi-Party Pose Estimation

Présentée le 26 novembre 2021

Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Angel Noe MARTINEZ GONZALEZ

Acceptée sur proposition du jury

Prof. J.-Ph. Thiran, président du jury
Dr J.-M. Odobez, directeur de thèse
Dr F. Moreno-Noguer, rapporteur
Dr O. Lanz, rapporteur
Dr A. Alahi, rapporteur

École
polytechnique
fédérale
de Lausanne

2021

TO MY PARENTS

Acknowledgements

First and foremost, I would like to thank to my supervisor Jean-Marc Odobez for provided me with his guidance, time and support. I have been fortunate to have crossed paths with him in Mexico and to have become a PhD student under his guidance. The experience of working with him has truly enriched my career.

I would like to thank Prof. Jean-Philippe Thiran, Dr. Alexandre Alahi, Dr. Francesc Moreno-Noguer, and Dr. Oswald Lanz for agreeing to be part of my jury and for their comments.

I acknowledge the financial support provided by the MuMMER project (funded by the European Commission) and the PhD scholarship program of the Mexican Council on Science and Technology (CONACYT).

I am truly grateful to my past and present colleagues in the Perception group at Idiap. Thanks to Olivier Canévet and Michael Villamizar for their invaluable help, insightful comments, and collaborations. Special thanks to Michael who without his guidance this thesis would not have been possible. A honorable mention goes to Dr. Johan Van Horebeek and Dr. Jean-Bernard Hayet from CIMAT who had always provided their true friendship and career support.

My sincere gratitude to all the wonderful people and friends during this journey. I will always cherish the moments we spend together for traveling, skiing & snowboarding, hiking, beers, coffee breaks, dinners, and so on. I thank Pierre-Édouard, Gulcan, Cijo, Pranay, Weipeng, James, Dhananjay, Apoorv, Sargam, Bastian, Sophie, François, Lesly, Michael, Hannah, FX, Clara, and the many more whose names are not here but were an important stone to make life at Martigny enjoyable. A special thanks with all my love to Rosa whose constant support and motivation has made everything look easier. Extra credits to Pierre-Édouard and Gulcan for adopting me during my internship at Google Zurich.

Finally, I would like to thank my parents for their unconditional love, patience, guidance, blessings, and encouragement they provided me to follow whatever interests me. Thanks to my brothers and sisters for their encouragement and love. A special thanks to my sister Nahara and her better-half Marc for all their kind help I received while living in Switzerland, the many travels we did together and their constant guidance.

Martigny, July 2, 2021 Angel

Abstract

Human detection and pose estimation are essential components for any artificial system responsive to the presence of humans and that react according to human-centered tasks. Robotic systems are typical examples, for which the body pose represents fine grained information useful to understand the behavior and activities of people, and interact with them. However, it is a challenging research topic with increasing difficulty given the unknown number of people in a usual scenario and factors like occlusions and sensing conditions. Current state-of-the-art methods have largely used deep Convolutional Neural Networks (CNN) to address the task. Traditionally, the selected CNNs are very deep and overparameterized, hence requiring large amounts of data to achieve good generalization and prevent overfitting. As a consequence, they are not straightforward to deploy in the low budget hardware typically available in practical applications such as HRI.

This thesis studies methods for efficient and reliable 2D and 3D human pose estimation using deep learning approaches. It investigates novel lightweight convolutional network architectures that achieve real-time performance in multi-person scenarios and explore knowledge distillation methods to boost the performance of these models while keeping their efficiency. Moreover, this thesis addresses the high cost of data collection with annotations that arises with our deep learning-based approaches by relying in a large scale dataset of synthetic images with high variability. Domain adaptation methods and data augmentation strategies are proposed to exploit the synthetic corpus in order to achieve good generalization in sensor data. Additionally, this dissertation studies human 3D motion prediction framed as a sequence-to-sequence problem. Non-autoregressive transformer neural networks are proposed to predict elements in parallel to avoid error propagation from predicted elements, observed in autoregressive methods, while at the same time being efficient.

Overall this thesis proposes different efficient and accurate deep learning solutions to design components of a human behaviour understanding system exploited in Human-Robot-Interaction (HRI) scenarios.

Keywords: human robot interaction, human behaviour understanding, deep learning, person detection, human pose estimation, domain adaptation, knowledge distillation, human motion prediction, transformer neural network.

Résumé

Détecter des personnes et estimer la pose de leur corps sont des composants essentiels de tout système artificiel nécessitant d'analyser les activités des humains. Les systèmes robotiques intéragissants avec l'homme sont un exemple type de tels systèmes, ou la pose du corps représente une information détaillée permettant de mieux appréhender le comportement humain et d'agir en conséquence. Néanmoins, extraire une telle information présente de nombreuses difficultés liées aux variabilités intrinsèques des données, telles que le nombre de personnes, les occlusions et les vues partielles du corps, les conditions d'acquisition et d'illumination, ainsi que les variabilités d'apparence (vêtements) et physiques (taille, corpulence, haute variation des degrés d'articulation du corps). Les méthodes récentes sont largement basées sur l'utilisation de réseaux neuronaux convolutifs profonds (CNN en Anglais), qui sont traditionnellement très profonds et sur-paramétrés, requérant ainsi de grandes quantités de données pour obtenir une bonne généralisation et éviter le sur-apprentissage, et ont ainsi un coût opératoire important. Cela rend leur utilisation difficile avec les resources de calcul généralement disponibles dans des applications pratiques telles que les systèmes embarqués en interaction homme-robot (HRI en anglais).

Cette thèse s'intéresse au design de méthodes efficaces pour l'estimation 2D et 3D de la pose du corps basées sur des approches d'apprentissage profond. Elle étudie l'usage de caméra de profondeur ainsi que des nouvelles architectures légères de réseaux convolutifs permettant d'obtenir des estimations en temps réel dans des scénarios. Elle explore également des méthodes de distillation des connaissances pour améliorer les performances de ces modèles tout en conservant leur efficacité. En outre, elle aborde le coût élevé de la collecte de données et de leurs annotations face aux besoins des approches en apprentissage profond et propose d'employer à grande échelle des images de synthèses dont l'annotation est ainsi obtenue de manière automatique. Des stratégies d'augmentation des données ainsi que des méthodes d'adaptation du domaine sont alors proposées afin de pouvoir bénéficier des ces données de synthèse tout en obtenant une bonne généralisation des capacités d'estimation avec des images de profondeur réelles. Enfin, cette thèse étudie la prédiction du mouvement humain en coordonnées 3D sous la forme d'un problème de prédiction séquence-à-séquence. Des réseaux neuronaux "transformers" sont proposés pour prédire en parallèle les différentes poses à différents horizion temporels et éviter les propagation d'erreurs observées avec les méthodes autorégressives traditionnelles.

De manière générale, cette thèse propose différentes solutions d'apprentissage profond performantes et précises de la pose utiles pour des systèmes de compréhension du comportement humain, comme ceux exploités dans le domaine de l'interaction homme-robot.

Mots clé : interaction homme-robot, analyse du comportement humain, apprentissage profond, détection de personnes, estimation de la pose du corps, adaptation de domaine, distillation des connaissances, prédiction du mouvement humain, réseau neuronal transformer.

Contents

Acknowledgements i

List of Tables ix

List of Figures xii

1 Introduction 1

1.1 Background . . . 2
1.2 Human Behaviour Understanding Pipeline . . . 3
1.3 Deep Sensing . . . 3
1.4 Challenges . . . 5
1.5 Scope of the Thesis and Contributions . . . 6
1.6 Thesis Outline . . . 8

2 Literature Review 11

2.1 Introduction and Problem Definition . . . 11
2.2 2D Human Pose Estimation . . . 12
2.2.1 Classical Machine Learning Approaches . . . 12
2.2.2 Deep Learning Approaches . . . 15
2.3 3D Human Pose Estimation . . . 18
2.3.1 Monocular 3D Pose Estimation . . . 19
2.3.2 3D Pose from Depth Images . . . 20
2.3.3 Multi-person 3D Pose Estimation . . . 22
2.4 3D Motion Prediction . . . 22
2.4.1 Deep Learning for Motion Prediction . . . 23
2.5 Domain Adaptation . . . 26
2.5.1 Learning from Synthetic Data . . . 27
2.5.2 Unsupervised Domain Adaptation . . . 27
2.5.3 Beyond image classification . . . 29
2.6 Model Compression and Knowledge Distillation . . . 29
2.6.1 Model Compression . . . 30
2.6.2 Knowledge Distillation . . . 31
2.7 Conclusion . . . 31

3 Depth Image Databases 35
3.1 Synthetic Depth Image Dataset 36
3.1.1 Synthesizing Images of People 36
3.1.2 Randomized Rendering Pipeline 36
3.1.3 S-DIH Dataset 39
3.1.4 Real Background Fusion 40
3.2 Real Depth Image Datasets 40
3.2.1 R-DIH Dataset 41
3.2.2 CMU Panoptic Dataset 41
3.2.3 ITOP Pose Estimation Dataset 42
3.3 Summary 43

4 Efficient 2D Multi-Person Pose Estimation 45
4.1 Efficient Human Pose Estimation 46
4.1.1 Pose Machines CNN Architecture Class 46
4.1.2 Efficient Pose Machines 47
4.1.3 Confidence Map Ground Truth and Training 50
4.1.4 Body Part Association 51
4.2 Distilling The Pose From a Network 51
4.2.1 Mimicking Teacher Stage Predictions 53
4.2.2 Learning With Hints 53
4.3 Experiments 54
4.3.1 Data 54
4.3.2 Evaluation metrics 55
4.3.3 Implementation Details 55
4.3.4 Performance-Efficiency Trade-Off 56
4.3.5 Training with Synthetic Data Analysis 60
4.3.6 Pose Knowledge Distillation 62
4.4 Summary 63

5 Domain Adaptation for Pose Estimation 65
5.1 Depth Image Domains 67
5.2 Unsupervised Adversarial Domain Adaptation 67
5.3 Other Adaptation Methods 69
5.4 Network Architectures 70
5.4.1 Domain Classifier 70
5.4.2 Pose Regressor CNN 70
5.4.3 Confidence Map Prediction and Pose Regression Loss 71
5.5 Experiments 72
5.5.1 Data 72
5.5.2 Evaluation Metrics 72
5.5.3 Implementation Details 72
5.5.4 Results 74

5.6 Summary 75

6 Multi-Person 3D Pose Estimation 77

6.1 Approach Overview 78

6.2 Efficient 2D Pose Estimation and Lifting 79

6.2.1 CNN-based 2D Pose Estimation 79

6.2.2 Pose Lifting 80

6.3 Human 3D Pose Estimation 81

6.3.1 Residual Pose Learning 81

6.3.2 Neural Network Regressor 82

6.3.3 Pose Learning Loss 82

6.4 Experiments 83

6.4.1 Depth Image Datasets 83

6.4.2 Evaluation Metrics 83

6.4.3 Implementation Details 84

6.4.4 Experimental Results 84

6.5 Summary 86

7 Human Motion Prediction 89

7.1 Approach Overview 90

7.2 Method 91

7.2.1 Problem Formulation 91

7.2.2 Pose Transformers 92

7.2.3 Pose Encoding and Decoding 93

7.2.4 Activity Recognition 94

7.2.5 Training 95

7.3 Experiments 96

7.3.1 Data 96

7.3.2 Implementation details 96

7.3.3 Evaluation metrics 98

7.3.4 Results 98

7.4 Summary 104

8 Conclusions and Future Directions 107

8.1 Summary of Contributions 107

8.2 Perspectives 108

Bibliography 111

Bibliography 121

Curriculum Vitae 123

List of Tables

3.1 Summary of the physical features contained in the 3D characters collection. . . 37
3.2 Summary of the number of images in the DIH dataset. 40

4.1 Performance (%) on the test set of real depth images and architecture components for the different tested pose machines instances. 56
4.2 Comparison of performance obtained with RPM-2S when trained with different synthetic data partitions. 60
4.3 Performance obtained with the RPM-2S model when combining synthetic data with only single person images (1P-Fold) and 2 people images (2P-Fold) for training. 61
4.4 Knowledge distillation experiments. Methods with - in the 'Teacher' and 'Distil type' fields indicate that the model has been trained without distillation and their values were taken from Table 4.1 (i.e. these are the baseline results). . . . 62

5.1 Comparison of performance obtained with the different techniques for domain adaptation using the RPM-2S model. CR stands for the confusion rate obtained in the domain classification task. Values with * were obtained by training a domain classifier using the corresponding model feature extractor as fixed features. Inpainting preprocessing corresponds to the application of an inpainting step on the real depth image before applying the detector. 73

6.1 2D and 3D pose estimation performance obtained for the different 2D CNN architectures and their computational complexity. 84
6.2 Visualization of 3D pose estimation performance. 87

7.1 **H3.6M** prediction performance. Top: autoregressive (POTR-AR) and non-autoregressive POTR models using linear layers for networks ϕ and ψ. Bottom: non-autoregressive POTR models with GCNs for network ϕ (enc), network ψ (dec) and both (full). Values correspond to the Euler angle error performance averaged over all actions. Lower values are better. 99
7.2 **H3.6M** comparison with the state-of-the-art for the common *walking, eating, smoking* and *discussion* for across different horizons. Values are the euler angle error. 99

7.3 Prediction results for the reminder of the 11 actions in the **H3.6M** dataset with our main non-autoregressive transformer. 100
7.4 **NTU** motion prediction performance on different time ranges with autoregressive and non-autoregresive POTR. Model marked with *memory* replace the class token with the encoded memory for activity classification. Values correspond to the mean average precision in a 10 cm range (mAP@10cm). Higher values are better. 101
7.5 Activity classification performance comparison with the state-of-the-art in the **NTU** dataset using different input modalities. 104

List of Figures

1.1 Multi-person Human-Robot Interaction scenarios. 2
1.2 Pipeline of a human behaviour understanding system. 3
1.3 Addressed elements of the human behaviour pipeline we investigate in this thesis. 6

2.1 Illustration of tree-like structure human pose priors. 12
2.2 Illustration of (Shotton et al., 2012) for pose estimation from depth images. . . 13
2.3 Illustration of classical methods for 2D pose estimation. 14
2.4 CNN-based cascade of detectors for body parts confidence map prediction. . . 15
2.5 Hourglass architecture. 16
2.6 Multi-person pose estimation with part-affinity fields. 17
2.7 Illustration of 3D pose estimation with missing landmarks. 18
2.8 (a) Offset regression voting for 3D pose prediction from depth surfaces. (b) Illustration of voxelized representation of a hand depth surface as computed in (Moon et al., 2018). 21
2.9 Simplified overview of human 3D motion prediction. 24
2.10 Illustration of (a) autoregressive and (b) non-autoregressive motion prediction. 24
2.11 Accuracy vs size comparison of state-of-the-art architectures in ImageNet challenge for image recognition (source (Canziani et al., 2016)). 30

3.1 (a) Sample 3D characters with different body features and outfits (b) 3D characters retargeted with different CMU-Mocap sequences. 36
3.2 Randomized image synthesis pipeline 37
3.3 Synthesis scenario configuration. 39
3.4 Examples of recorded real data in HRI scenarios 42
3.5 Examples of the CMU-Panoptic dataset. 43
3.6 Examples of the ITOP dataset. 43

4.1 Pose machine architecture class. 46
4.2 CNN architecture instances of the pose machines class. 48
4.3 Different unit modules used in our architecture designs. 50
4.4 Knowledge distillation for body landmark detection. 52
4.5 Precision-recall curves obtained by varying the landmark detection threshold η. 57
4.6 Precision (left) and recall (right) per body landmark for the RPM-3S and the baseline CPM-2S. 57

4.7 Visualization examples of pose estimates in DIH and CMU-Panoptic datasets. . 58
4.8 Left: efficiency-accuracy tradeoff visualization for the best models on DIH-Real. Models marked with * have been trained with knowledge distillation. Right: evaluation of the use of synthetic data for learning robust pose estimation models. 60

5.1 Scheme for pose estimation learning and unsupervised domain adaptation. . . 66
5.2 Depth imaging characteristics of different sensors. 67
5.3 CNN architectures for adversarial domain adaptation. 69
5.4 Comparison of different techniques for synthetic-to-real domain adaptation. . 73

6.1 Overview of our decoupled residual pose approach. 78
6.2 CNN architectures used for 2D pose estimation. 79
6.3 Skeleton and limb pairwise relationships. 80
6.4 Illustration of the calibration error introduced by the lifting 2D detected landmarks. 81
6.5 Residual pose learning framework. 82
6.6 Performance of the different CNNs for 2D pose estimation. 85
6.7 3D pose estimation examples and their 2D projection of our approach on the single person ITOP dataset (top row) and the multi-person CMU-Panoptic dataset (bottom row). 86

7.1 Proposed motion prediction approach. A transformer with non-autoregressive decoding predicts future motion from a generated sequence composed from a query pose. 90
7.2 Human motion prediction with non-autoregressive transformers 92
7.3 Encoder and decoder architectures each with L layers. As in (Vaswani et al., 2017), each layer is composed of feed forward networks and multi-head attention layers. 94
7.4 Overview of our Graph Convolutional Network architecture. It is composed of graph convolition layers followed by *tanh* activations, batch normalization, and dropout layers. Similar to (Wei et al., 2020), our architecture comprises S residual connections. In practice we used $S = 1$. 95
7.5 Examples of images from: (a) H3.6M dataset, and (b) the NTU action dataset. . 97
7.6 Qualitative results for the H36M dataset. 101
7.7 **H3.6M** datasest encoder-decoder attention weight visualization. 102
7.8 NTU per body part motion prediction performance in terms of (a) mAP@10cm (the higher the better) and, (b) MPJPE (the lower the better). 103
7.9 Confusion matrix for the NTU dataset. 105

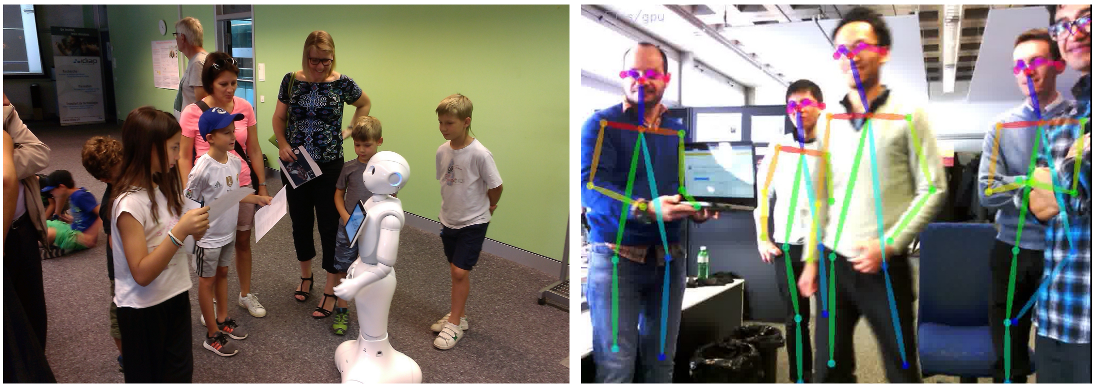
1 Introduction

In the past decades Artificial Intelligence has striven to build systems that are capable to perceive its environment and to react according to human-centered tasks. For example, a health care system may have the capability to detect when a patient has fallen to the ground and to emit an alarm to require medical attention. These systems are responsive to the presence of humans and provide support for decision making on scenarios involving people's activities. As such, one of their main goals is to understand human behaviour from sensory data, e.g. video, audio, etc. Their functionality relies in a perception module that takes the sensed information and processes it for effective human behaviour understanding.

Visual detection and tracking of people are core components of any artificial system aware of human presence. These provide a history of people's whereabouts and body motion which are later used for interpreting intentions. We humans commonly understand people's behaviour by inspecting their motion, body gestures, and other non-verbal information in a given scene context. However, this cannot be easily achieved by artificial systems. For these, obtaining such high-level understanding requires a process in which low-level image signal information is first extracted, then used to generate mid-level motion representations to finally process them for interpretation.

The work presented in this thesis addresses the tasks of visual detection and motion modelling of people for *Human-Robot Interaction* (HRI) settings. Our work studies several deep learning approaches for efficient and reliable visual detection of people and body pose estimation. Opposite to other domains requiring human behaviour interpretation such as autonomous driving or visual surveillance, in HRI scenarios people interact directly with a robotic platform at different distances and directions (see Figure1.1). Therefore, localizing people in the surroundings of the robot and modelling their motion is essential for successful interpretation of their interactions.

The rest of this chapter presents an overview of our research. First, we introduce the context in which the thesis research was conducted. Then, we introduce the common pipeline for human behaviour understanding identifying the components investigated in this thesis. Subsequently,



(a) (b)

Figure 1.1 – (a) Typical HRI scenario where the robot interacts with multiple people; (b) Example of 2D multi-person pose estimation on an RGB image.

we frame our research within the deep learning-based methods and discuss its advantages over classical machine learning. We introduce various challenges encountered in HRI scenarios and based on these we set the scope and objectives of our work. Finally, we introduce the contributions of this thesis.

1.1 Background

The research in this thesis was conducted within the European project *MuMMER* [1] (*Multimodal Mall Entertainment Robot*). Its goal is to build social intelligent robots for entertainment in public spaces (Foster et al., 2019, 2016).

Pepper[2] from *Softbank Robotics* has been used as the robotic platform. The robot is equipped with two RGB cameras (OV5640 5MP) and one depth camera (Asus Xtion 3D) among other sensors (microphones, and contact). Potentially, it can be equipped with other cameras (e.g. Kinect 2) to increase the robot field of view.

The robot is expected to perform in unconstrained HRI interaction scenarios, like the one shown in Figure 1.1(a). Such interactions involve answering questions, providing instructions, telling jokes, etc. Though the scenario conditions are mostly indoors, the number of people interacting with the robot is unconstrained as well as other scenario conditions like illumination, background clutter, person occlusions, etc. The interactions should be handled as natural as possible, which means that all the modules of the robot need to operate in real time.

---

[1] http://mummer-project.eu/

[2] https://www.softbankrobotics.com/emea/en/pepper

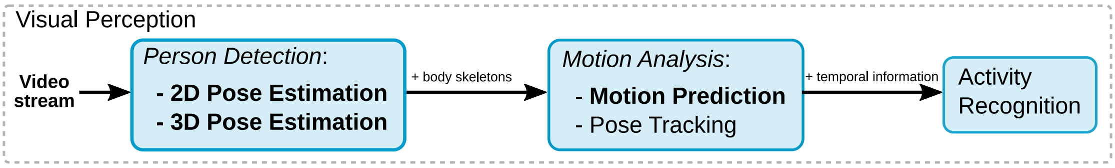


Figure 1.2 – Pipeline of a human behaviour understanding system from visual modality. We highlighted the focus of this thesis in bold fonts. Note that not all modules are time-dependent.

1.2 Human Behaviour Understanding Pipeline

An artificial system designed for human behaviour understanding implements *Human-Sensing* (Teixeira et al., 2010) aiming to answer questions such as: how many people are in the room? or, what activity is being performed? It encompasses the challenges from low-level signal processing and detection to high-level information modeling for interpretation, including the analysis of group activities (Reiter-Palmon et al., 2017). Understanding human behaviour, and communication cues in particular (like detecting the speaking status (He et al., 2018; He et al., 2021), nods (Nguyen et al., 2012; Chen et al., 2015), or attention (Ba and Odobez, 2006; Sheikhi and Odobez, 2015; Siegfried et al., 2017)), is of great importance in HRI settings in order for the robot to respond to interactions in a proper way.

Although a system can incorporate information from different sensor modalities, i.e. video and audio, our interest resides on the video modality. Using video presents several advantages over other sensor modalities. First, non-verbal communication in the form of body gestures (Wu et al., 2016) or visual focus of attention (VFOA) can be extracted from images. It also promotes the extraction of scene image context, e.g. identifying the objects a person is interacting with. Finally, access to the 3D world is provided by depth camera sensors and foster 3D scene understanding.

A common workflow for human behavior understanding involves many technologies, from person detection to activity recognition. These normally work in a hierarchical fashion, thought not always sequentially. Figure 1.2 shows a diagram of its core components. It works as follows: a) a stream of images is acquired from the camera sensor and presented to the pose detection module; b) people are detected in a per-image basis and their individual body pose is extracted, in image (2D) or world (3D) coordinates; c) spatio-temporal properties of the body motion are retrieved in order to perform motion prediction and pose tracking; d) finally, high level understanding such as activity recognition is performed from motion cues.

1.3 Deep Sensing

In recent years deep learning techniques have become the leading algorithms for many machine learning tasks, from object detection (Redmon et al., 2016) in computer vision, medical applications (Ciller et al., 2017) to machine translation (Vaswani et al., 2017). All this has been made possible thanks to the increasing number of large available datasets with annotations (Lin et al., 2014; Wang et al., 2018), increasing computational power with specialized accelerators (i.e. GPUs or TPUs), and large improvements in optimization like tricks to prevent from vanishing gradients.

As with many other fields in computer vision, the use of deep learning algorithms have replaced classic machine learning for human sensing related tasks. Very deep models have shown drastic performance increases in topics from 2D body pose estimation (Cao et al., 2017), crowd behaviour analysis (Kothari et al., 2021) to video activity captioning (Sudhakaran et al., 2021; Wang et al., 2018). These tasks profit from several advantages of deep learning methods over classical ones (LeCun et al., 2015):

- **Representation learning approach**. Representation learning allows a machine to automatically discover adequate representations from raw data. In classical approaches these features were manually designed and required some level of expertise. For instance, deep learning models for 2D body pose estimation leave out the classical explicit hand-design of prior models of the kinematics of the body and instead learn these implicitly from examples. In practice, a *Deep Neural Network* (DNN) can learn meaningful representations from the training data as long as enough quantities of representative examples and sufficient DNN learning capacity (number of neurons, layers, etc.) are provided.
- **Locality exploitation**. For signals such as images, pixels in a neighborhood are highly correlated with statistical properties that are invariant to location. *Convolutional Neural Networks* (CNN) take advantage of these properties to detect translation invariant patterns and merge similar semantic features (e.g. with pooling). As such, person detection and pose estimation methods are robust in very challenging scenarios under different sensing conditions (illumination, noise, etc) and multi-person settings with partial observations and scales. For a classical approach it is hard to achieve feature selection that helps to obtain more discriminative feature representation.
- **Hierarchical composition**. DNNs exploit the compositional semantic meaning of signals where high-level properties are composed of low-level ones. For example, in images local patterns (edges, corners, colors, etc.) can be assembled into body parts, and body parts into a human body. DNNs achieve different levels of feature representation by composing (stacking) many non linear modules (layers) that convert features to abstract levels. The deeper the composition, the more complex the function that can be learned. Specific to the body pose estimation literature, the trend is to design very deep CNNs to localize body landmarks in the image and sequentially refine these locations with staked sets of layers. Such compositional property allows these deep models to be learnt end-to-end even in multi-task settings.

In this thesis we adopt deep learning techniques for the different elements of the human behavior understanding pipeline. Our research centers to the analysis of the human body from images in multi-person settings, like the one shown in Figure 1.1(b). However, regardless their breathtaking results, deep learning methods bring different challenges to specific targeted scenarios. We introduce these shortcomings for our HRI scenario in the next section.

1.4 Challenges

Though in recent years deep learning techniques have made substantial progress in person detection, human pose estimation and person tracking, their practical application in HRI scenarios is not evident. On one hand, the dynamic nature of HRI scenarios, the background clutter and sensing conditions may lead to partial occlusions hence hindering perception. On the other hand, accurate but more complex systems may bring high computation burden, disabling the possibility for real-time performance under the limited computational budget of the robotic platform. More precisely, we identify the following challenges:

1. **Unconstrained scenarios and variability**. Dynamic scenarios may contain an unknown number of people. For HRI people normally interact naturally with the robot, between each other, and with the objects in the environment through body gestures or other non-verbal communication. In this setup, factors such as robot's motion, background clutter as well as between-person interactions may provoke partial observations and often detection.

2. **Real-time performance requirements**. Robotic platforms are normally equipped with lightweight CPU or GPU processors with very few resources for data processing. Moreover, these resources have to be shared among different modules, like motion planning, localization, dialog managing, etc. Multi-person detection and tracking for HRI systems require real-time running performance to achieve fluent interactions. Yet, accurate DNNs models tend to be very deep and large in terms of number of parameters. This brings high computational demands, normally requiring GPU processors to deploy the models (sometimes not even in real time).

3. **Data needs for training.** DNNs require large amounts of data with enough variability and high-quality ground truth for the learning phase. Existing benchmark datasets might not meet these requirements for specific scenarios like those encountered in HRI. Hence, obtaining or collecting such datasets for our particular application can be very expensive and time consuming.

We address these challenges with different deep learning techniques to find a good trade-off between efficiency and performance. In our research we do not make strong assumptions about the scenario, e.g. number of people, body pose, or close sensing range. However, thoughout this thesis most of the experiments were conducted on low budget GPU accelerator.

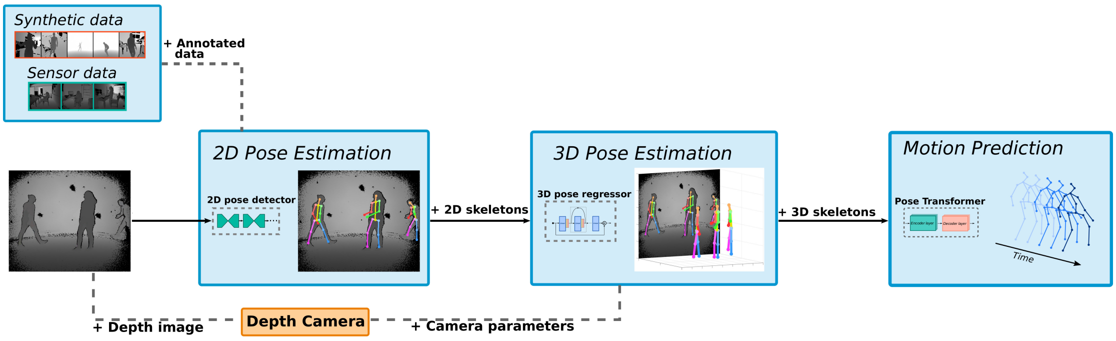


Figure 1.3 – Addressed elements of the human behaviour pipeline we investigate in this thesis. (a) we synthetsize and collect depth image databases for training; (b) we investigate efficient methods for 2D pose estimation; (c) we leverage on fast 2D pose estimation for 3D pose regression; (d) we introduce a method for 3D motion prediction.

1.5 Scope of the Thesis and Contributions

The overall goal of this thesis is to investigate and develop practical deep learning methods for human behaviour analysis systems in HRI scenarios and more specifically methods related to human body pose analysis. In this context, we focused in the following tasks:

- Synthetic data generation for 2D human pose estimation;
- Efficient human 2D pose estimation;
- Efficient human 3D pose estimation;
- Human motion modelling and prediction.

As previously discussed, localizing body landmarks of people, in 2D and 3D provides the robot the means for fine-grained motion understanding and human activity recognition for social interactions. We operate with streams of depth images. Regardless of being simpler than color images, i.e. they have less diversity of color and texture, they contain rich and sufficient information for our tasks. They may thus necessitate less complex models to accomplish our objectives with a good trade-off between accuracy and speed. Additionally, they provide easy access to the 3D world easing the extraction of 3D motion and fostering 3D scene understanding.

Based on the previously discussed challenges, the more technical objectives of this thesis include:

- Investigate tools for data generation to cover the need for training data of deep learning approaches.
- Investigate efficient CNN architectures and deep learning techniques for fast and reliable multi-person pose estimation, both in 2D and 3D

- Investigate deep learning domain adaptation techniques to transfer the learned capabilities from synthetic data to real sensor data.
- Investigate deep learning approaches for human motion modelling and prediction.

Inline with our objectives we worked on the different modules illustrated in Figure 1.3. These can be seen as implementations of the elements of the human behaviour understanding pipeline shown in Figure 1.2 Concretely, the main contributions of this thesis can be summarized as follows:

- **A dataset of depth images for human 2D pose estimation**. To cover the need for training data, we worked with computer graphics and 3D character design tools to generate synthetic images. The result is a large scale dataset of depth images for 2D human pose estimation, including more than 260K synthetic depth images with 2D and 3D landmark annotations, and more than 15K Kinect 2 depth images of sequences of multi-person HRI scenarios with a real robot. Our dataset dubbed as DIH (Martínez-González et al., 2018b) has been made public for research and commercial purposes [3]. Additionally, the code of our tool for data synthesis has also been made public [4].

- **Fast 2D body landmark localization from depth images**. We proposed several efficient and novel lightweight CNNs models for fast and reliable 2D human pose estimation comprising a cascade of detectors. The design of these architectures exploit modules from ResNets, MobileNets and SqueezeNets to reduce the model's size and computation requirements of convolutions. Additionally, we explore knowledge distillation techniques to boost the generalization capabilities of these lightweight models coupling the distillation at different parts of the architectures to match our cascade of detectors approach. Our networks match our speed and accuracy requirements for real-time processing and outperforms state-of-the-art models. Part of this work has been published in (Martínez-González et al., 2020a) and (Martínez-González et al., 2018b); code has been made public public [5]

- **Domain adaptation**. We studied several deep learning domain adaptation methods for multi-person pose estimation, training DNNs with depth synthetic data, and exploiting unannotated real data for adaptation. We investigated unsupervised adversarial domain adaptation (Ganin et al., 2016). Additionally, we investigated approaches that apply specific data transformations during training or testing: include depth sensor information or making the testing images to look like synthetic ones. This study highlights domain adaptation limitations and the importance of including sensor information during training with synthetic data. Part of this work has been published in (Martínez-González et al., 2020a) and (Martínez-González et al., 2018a).

---

[3] https://www.idiap.ch/dataset/dih

[4] https://github.com/idiap/depth_human_synthesis

[5] https://github.com/idiap/fast_pose_machines

- **3D pose estimation**. We investigated a novel method for decoupling the 3D pose estimation task into an accurate and efficient CNN-based 2D bottom-up multi-person pose estimation method and 3D pose regression. Our method exploits the depth information and uses a simple 2D-to-3D lifting scheme which handles 2D body joint miss detections. We introduce a novel method for 3D pose regression from lifted 2D estimates by relying on a residual-pose deep-learning architecture. This approach, despite its simplicity, achieves very competitive results on different public datasets and is suitable for real-time multi-party HRI scenarios. This work was published in (Martínez-González et al., 2020b) and code and models are available [6].

- **Human 3D motion prediction**. We investigated a non-autoregressive approach for 3D human motion modelling and prediction. We addressed the problem as sequence-to-sequence prediction and rely in a transformer neural network to model the temporal dependencies between past and future 3D human pose sequences. Compared to autoregressive prediction, our approach uses the decoding mechanism of transformers in a non-autoregressive setting to avoid error propagation and reduce computation costs at testing time. We explore different network architectures to compute single 3D pose embeddings to model body spatial dependencies such as Graph Convolutional Networks (GCN). Additionally, we exploit the transformer encoder self-attention embeddings to perform activity classification. Our approach achieves competitive results in motion prediction benchmark datasets. This work was published in (Martínez-González et al., 2021) and code and models are available [7].

1.6 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 summarizes previous work related to 2D and 3D human pose estimation, motion prediction and different deep leaning topics related to our work.

Chapter 3 introduces the datasets we use throughout our research. First, we describe our randomize synthesis pipeline protocol to generate synthetic data using 3D human characters and motion simulation. Then, we introduce our protocol for data collection of people interactions with a robotic platform in HRI settings. Finally, we present other public benchmark datasets used in our experiments.

Chapter 4 introduces our work for efficient 2D multi-person pose estimation from depth images. We present our novel lightweight CNN architectures designed to achieve real-time performance. Additionally, we introduce our research for knowledge distillation to boost the performance of our lightweight architectures while keeping their efficiency.

---

[6] https://github.com/idiap/residual_pose

[7] https://github.com/idiap/potr

In Chapter 5 we describe our research on domain adaptation for learning 2D pose estimation for learning with synthetic data to perform in sensor data. Our research investigates unsupervised adversarial domain adaptation as well as data transformations applied to training and testing images, and highlights their limitations

Chapter 6 introduces our method for multi-person human 3D pose estimation from depth images. We detail our method for lifting and recover from 2D failures and our residual pose learning to predict offsets from 3D lifted skeletons.

Chapter 7 addresses 3D human motion prediction leveraging in transformer neural network architectures. We propose a non-autoregressive transformer that works on sequences of 3D skeletons to model the temporal dependencies for future motion sequence prediction and extract information relevant for activity recognition.

Finally, Chapter 8 presents our conclusions of the thesis summarizing our work and presenting suggested future directions that can build upon this work.

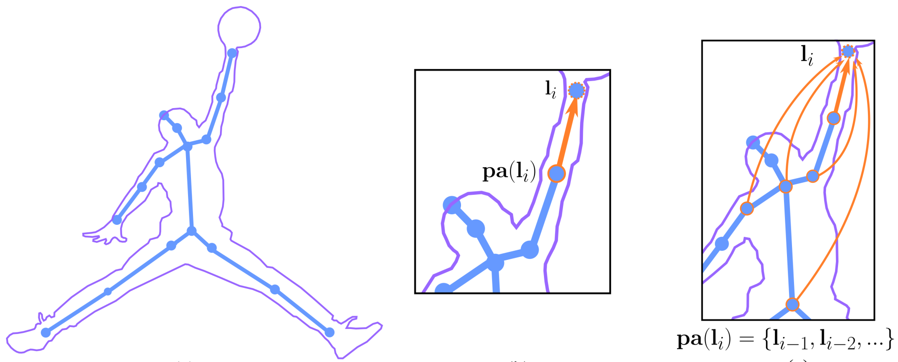
2 Literature Review

2.1 Introduction and Problem Definition

Human pose estimation is the task of predicting the body structure by means of detecting anatomical landmarks (body parts) in images, i.e. head, neck, hands, etc. Given its vast number of applications, such as visual surveillance, gaming, HRI, it has been a computer vision subject studied for decades. However, the large complexity of the body structure makes the task very challenging: the human body, as an articulated object, has a wide range of pose configurations and people may appear at different profiles to the camera, occluding parts of the body. Additionally, human motion is highly non-linear with large uncertainty due to the articulated structure of the human body.

Research in human pose estimation and motion prediction has addressed the problem of estimating the human pose in the 2D (image coordinates) and the 3D spaces (a 3 dimensional coordinate frame). Regardless it is 2D or 3D space, the problem starts on training a set of body part detectors that localizes them in the image plane. For the 3D case, 2D pose estimation is usually an intermediate process. However, depending on the input modality, estimating the 3D pose often requires multi-camera settings or prior knowledge to model anatomical constraints. Though 2D coordinates is the most common representation used, some anatomical properties are not well represented. One of such challenges is inferring the locations of the self-occluded landmarks from 2D information only. Additionally, human motion can be better understood in the 3D space where prior information about the anatomical properties, such as body proportions and articulation limits of the body are better represented.

This chapter reviews research related to the components of the human behaviour understanding pipeline for HRI scenarios. We limit the scope to the topics most relevant to this thesis work, i.e. human 2D and 3D pose estimation, human motion prediction, domain adaptation and knowledge distillation.


$\mathbf{l}_i$
$\mathbf{pa}(\mathbf{l}_i)$
$\mathbf{l}_i$
$\mathbf{pa}(\mathbf{l}_i) = \{\mathbf{l}_{i-1}, \mathbf{l}_{i-2}, ...\}$

(a) (b) (c)

Figure 2.1 – Illustration of tree-like structure human pose priors. (a) Simple human body tree-like body representation; (b) body part pairwise relationships where limb $\mathbf{l}_i$ has a single element in its parent set $\mathbf{pa}(\mathbf{l}_i)$. (c) denser body parts relationships with limb $\mathbf{l}_i$ with multiple parents in its parent set $\mathbf{pa}(\mathbf{l}_i)$.

2.2 2D Human Pose Estimation

In this section, we summarize methods for 2D human pose estimation. We start by discussing classical machine learning approaches and then introduce more recent deep learning based ones.

2.2.1 Classical Machine Learning Approaches

Classical machine learning methods focused on solving the single-person pose estimation problem from person centered image crops. These methods focused on solving body part detection together with generating estimates with physically plausible body pose configurations. To detect human body parts, it is usual to employ per body part specialized detectors trained on features manually designed by experts. Consequently, the detected body parts are coupled together taking into account the body structure kinematics as constraints to generate feasible estimates.

One of the most widely used practices for 2D pose estimation is the *Pictorial Structures* model (Yang and Ramanan, 2013; Andriluka et al., 2009; Johnson and Everingham, 2010). Such approach couples 2D detected body parts considering the human body as a tree-like structure. Body parts correspond to the nodes while the limbs connecting them, e.g. forearm, arm, spine, etc., are modelled as edges. Body parts relationships are usually designed in pairs to ease the pose inference stage. See Figure 2.1. Other non-tree-based approaches instead aggregate context from all the body part detectors to model body part relationships (Ramakrishna et al., 2014). They work as a message passing mechanism where all the body part detectors have the opportunity to incorporate context about the location of other body parts in the image, hence the relationships are learn implicitly rather than explicitly.

Depth data has also been used for human pose estimation related tasks (Shotton et al., 2012;

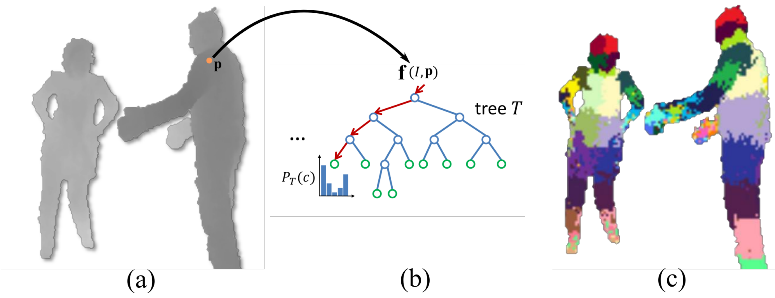


Figure 2.2 – Illustration of (Shotton et al., 2012) for pose estimation from depth images. (a) Depth image after background substraction; (b) pixelwise classification of human body parts with random forests; (c) pixelwise labels after classification.

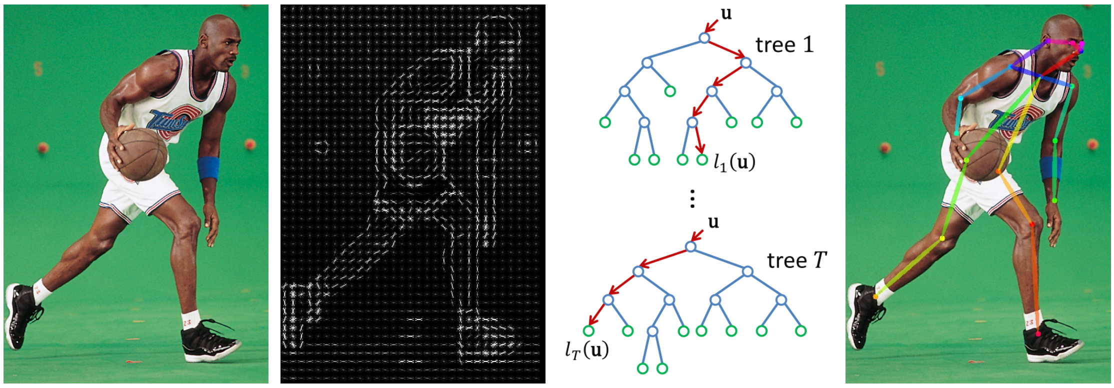
Jung et al., 2015; Kim et al., 2015). Contrary to color images, depth image values contain the distance from objects to the depth camera. They are relatively simpler: it contains mainly shape information in form of blobs of objects. However, they preserve many essential features that appear in natural images, e.g. corners and edges, and contain rich information about the 3D object's surface which might help to remove ambiguities in scale and shapes. In fact, they provide direct access to the 3D world which makes them very appealing for HRI applications.

The most well known approach employing depth images is the seminal work of Shotton et al. (Shotton et al., 2012). It follows the per body part detection setting. First, they preprocess the depth image by applying background subtraction assuming that the people are in a close range (5 meters). Pose estimation works in a pixel-wise classification setting using random forest, where each pixel in the depth image is classified as belonging to one of the different body parts of the body, i.e. head, shoulders, hands, etc. The combination of random forest and depth images lead to large improvements in both inference speed and estimation performance. Figure 2.2 illustrates this method.

Body Part Detectors and Features

A body part detector is a predictor that is trained to discriminate a given body part, e.g. head, from the background and other body parts, e.g. neck and shoulders. The predictor receives as input a vector of features $\mathbf{u}$ and provides a score related to the presence or the absence of the body part of interest. For a human body with N body parts, like head, neck or hands, there are N body part detectors each specialized in detecting the corresponding body part in the image.

The success of earlier pose estimation approaches relies in designing good features. These usually required expert knowledge and are engineered to enhance particular aspects of the image in order to be discriminative for the problem. In the pose estimation literature, one of the most used features for color images are the *Histograms of Oriented Gradients* (Dalal and Triggs, 2005). See Figure 2.3. In the case of depth images, (Shotton et al., 2012) used simple depth values comparison between a pixel $\mathbf{p}$ and two pixel offsets $\mathbf{p}_1$ and $\mathbf{p}_2$ which combined



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

Figure 2.3 – Illustration of classical methods for 2D pose estimation. (a) Input image tight crop; (b) HOG feature extraction; (c) Per body part classification with T random trees; (d) Predicted 2D pose.

with random forest lead to very fast inference.

Body part detectors operate from the set of extracted features to find instances of body parts. An abundance of works exist in the literature using *Support Vector Machines* (SVM) (Kim et al., 2015), random forest (Shotton et al., 2012, 2011), and boosted classifiers with trees as weak learners (Ramakrishna et al., 2014).

Anatomical Priors

To ensure a reliable pose detection of people in different and feasible body configurations, methods usually include information about the relationship between the different body parts of the human body. For example, leg extremities have articulation limits and cannot be directly connected to the head.

Most of the techniques relied in a Bayesian approach to model the body parts relationships and couple them with body part detections. For example, in the pictorial structures approach the human body structure L is modeled as the posterior probability $p(L|D) \propto p(D|L)p(L)$, where $P(D|L)$ is the image evidence computed with body part detectors, and $p(L)$ corresponds to a kinematic tree prior that models the relationship between landmarks

$$p(L) = \prod_i p(\mathbf{l}_i|\mathbf{pa}(\mathbf{l}_i)), \tag{2.1}$$

where the potential $p(\mathbf{l}_i|\mathbf{pa}(\mathbf{l}_i))$ model interactions between body landmarks and are learned from training data. In practical applications, the potentials are often pairwise modelled and designed using Gaussian likelihoods to ease inference and reduce computation time.

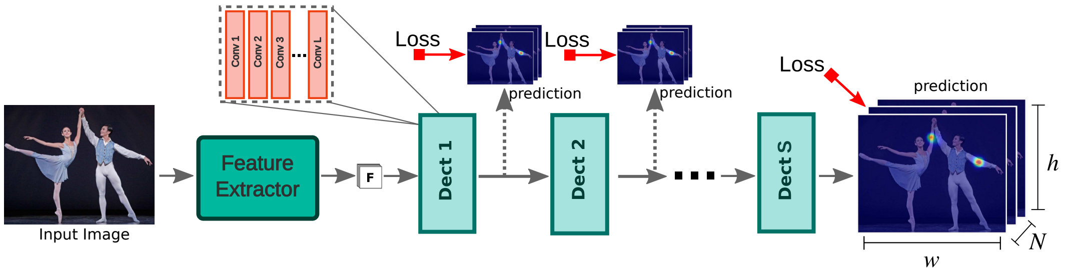


Figure 2.4 – CNN-based cascade of detectors for body parts confidence map prediction of N body parts. The feature extractor computes features **F** from the input image. Then a stack of detectors, each composed by convolutional layers, predicts and refines confidence maps of body parts in the image. This practice allows to incorporate training losses at every detector to boost performance and avoid vanishing gradients during training.

This however come at the cost of lower expressiveness. Other techniques overcome this by incorporating a more dense connection between body parts in the graph and perform inference with a message passing mechanism (Ramakrishna et al., 2014). See Figure 2.1(b) and (c) for an example.

2.2.2 Deep Learning Approaches

Recently, as with other computer vision tasks, CNNs have become the dominant approach for 2D pose estimation. The key advantage of deep learning methods over classical ones is that features are automatically learned using backpropagation, removing the feature engineering step in pursue for a more suitable image representation. These approaches also leverage the creation of a large number databases with body landmark annotations (Andriluka et al., 2014; Lin et al., 2014).

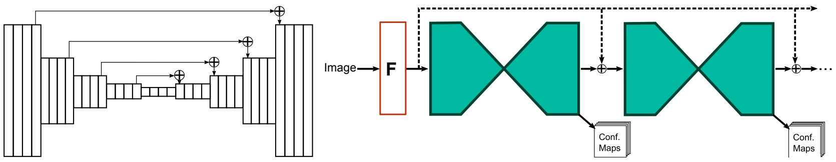
Deep learning methods for pose estimation often address the task as a cascade of body parts detectors, i.e. a stack of CNN detectors that refine predictions. They form deep architectures that regress per body part confidence maps encoding the per-pixel likelihood of body parts' positions. Figure 2.4 illustrates such a typical architecture. First, the input image is processed by a feature extractor, composed of several convolutional layers, that computes a representation of the image. Subsequently, these features are processed by a cascade of detectors that predict part confidence maps. Body part detectors are composed of a set of L convolutional layers receiving as input a tensor of shape $C \times H \times W$ and predicting as output a tensor of body part confidence maps of shape $N \times h \times w$, where N is the number of total body parts, and h and w are the resolution of the confidence maps. Finally, 2D location of body parts are obtained by selecting the pixel coordinates with maximum confidence.

The composition of the architecture, i.e. number of layers (depth), kernel size, and sequential pipeline, are designed in order to exploit different aspects of the image relevant to the task. For example, the architecture can be designed to predict more than one semantic output, e.g. body parts and limbs, in a multi-task learning approach where one task profits from the context of the other. The composition property of deep learning models allows to learn very deep models end-to-end with complex pipelines via backpropagation. In theory, the deeper the architecture the better the quality of the predictions. However, deeper models means more computational requirements and might nevertheless suffer from optimization problems.

In the remaining of this section we introduce recent methods for single and multi-person settings, for both RGB and depth image modalities.

Single-Person Pose Estimation

As in classical human pose estimation, early deep learning approaches assumed image tight crops of a single person and predict location of body parts. For example, the work presented in (Toshev and Szegedy, 2014) uses a deep CNN to regress and refine normalized (x, y) image coordinates of body parts. This practice is however difficult to optimize as it is highly non-linear in the prediction layers and shows a tendency to generalize poorly. The most extensively used convention is to use a deep CNN to output body part confidence maps as it has shown to have better generalization and sub-pixel accuracy (Wei et al., 2016).



(a) (b)

Figure 2.5 – Hourglass archictecture. (a) Hourglass module with downsampling, upsampling and skip connections. (b) Stack of hourglass for human pose prediction.

A plethora of approaches have emerged following these techniques proposing a very broad set of CNN architectures exploiting hierarchical features (Wei et al., 2016; Wang et al., 2016) or combining features at different semantic levels from inner layers (Newell et al., 2016; Hu and Ramanan, 2016; Yang et al., 2017). These early approaches modeled body part relationships by introducing context with large convolution kernels and very deep architectures. One of the most commonly used architecture is the *HourGlass* network (Newell et al., 2016). It comprises a cascade of UNet-like (Ronneberger et al., 2015) detectors as an encoder-decoder architecture. Each detector constitute a series of downsampling, upsampling and skip connections allowing to capture both contextual (arising from the lower resolution features) and local information (introduced by the skip connections). This upsampling strategy allows the architecture to learn to predict confidence maps at larger resolutions, obtaining good precision while being relatively efficient.

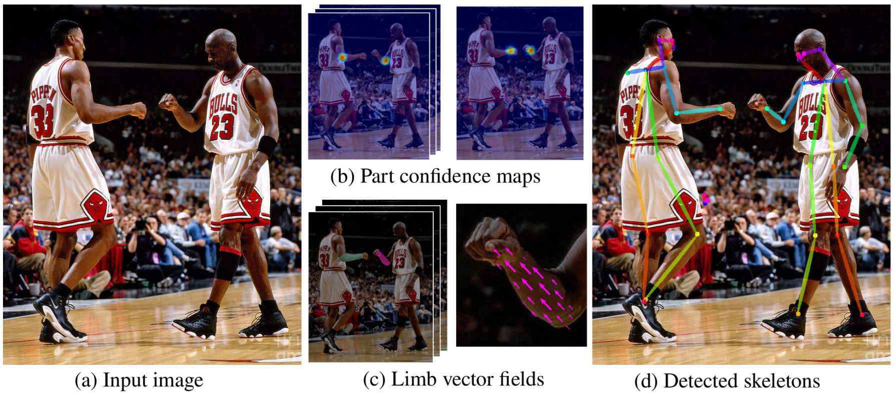


Figure 2.6 – Illustration of multi-person pose estimation with part affinity fields (limb vector fields)for body part association.

Multi-Person Pose Estimation

From a methodological perspective, multi-person pose estimation methods can be organized into top-down and bottom-up approaches. Top-down techniques first use a person detector (e.g. (Redmon et al., 2016)) to localize people in the image and then predict body landmarks for each of the individual detections (Chen et al., 2018). This approach leverages existing single-person techniques for the multi-person scenario. Conversely, bottom-up techniques first employ body part detectors to localize anatomical landmarks in the image (Insafutdinov et al., 2016). Subsequently, these detections are associated to build detections of individuals by exploiting a body structure connectivity (e.g. a tree). Despite the excellent results of top-down techniques, they are generally slow given that the computational cost grows with the number of detected people. We focus this review on bottom-up approaches.

Besides the body landmark detection of individuals, an important challenge in bottom-up methods is to perform the pose inference, i.e. the generation of person skeleton from the pool of detected body landmarks. Considering all possible combinations of detected body parts is non-feasible, and even a small amount of combinations is very time consuming (Pishchulin et al., 2016). To improve body part association and also reduce inference time, methods often introduce explicit spatial relationships between pairs of body parts in the CNN architecture and rely on multi-task learning for jointly infer body parts and their association (Cao et al., 2017; Fabbri et al., 2018; Sun et al., 2019).

The work presented by Cao et al. (Cao et al., 2017) has successfully applied the concept of cascade of detectors to build a CNN architecture for multi-person pose estimation with real-time performance. Figure 2.6 presents an overview of the pipeline. Their model first extracts features from the image with a deep feature extractor and then relies on a sequence of stacked layers to refine predictions. All the detectors in the cascade predict gaussian confidence for the location of body parts and vector fields encoding body parts pairwise dependencies. Such dependencies provide both the location and orientation of the body limb connecting two

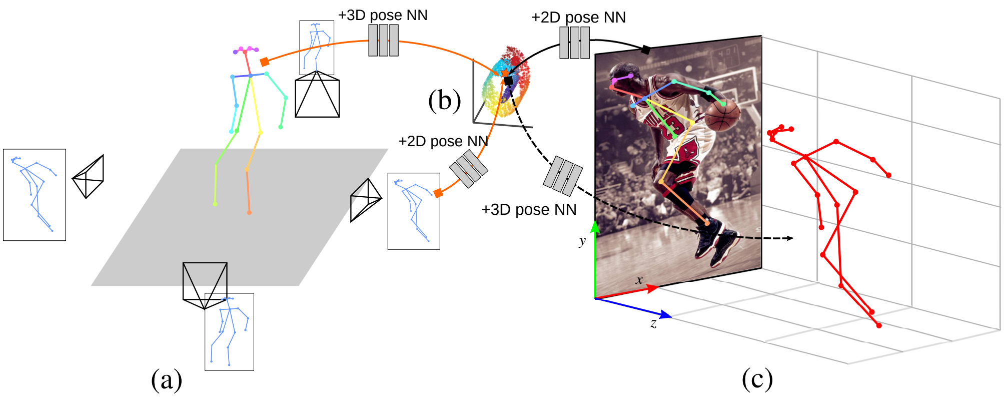


Figure 2.7 – Illustration of 3D pose estimation with missing landmarks as in (Li et al., 2015). Pairs of 3D and 2D body pose (a) are mapped to a joint embedding (b) with neural networks. (c) 3D pose is estimated from 2D pose with self-occlusions by mapping it to the joint embedding.

body parts, e.g. forearm is connecting the elbow to the wrist. To generate pose estimates the approach solves a bipartite matching problem considering only pairs of body parts connected in a tree-like skeleton. The method takes as input a given limb type, e.g. forearm. All detected body landmarks that form such limb type (wrists and elbows) are connected one to one, forming potential limbs. The connections are weighted by averaging the predicted vector field along the line that joins the pair of body landmarks. Connections with high confidence are kept while the others are discarded.

2.3 3D Human Pose Estimation

While there has been methods aiming at solving partial pose estimation, like extracting the body and head orientation for social behavior analysis in surveillance scenarios (Chen et al., 2011a), in general 3D pose estimation methods aim to estimate the full 3D coordinates of body parts of people appearing in an image, in a given coordinate reference system. Solving the problem usually consist of finding the 3D pose that better fits both the physical properties of the human body (*anthropometry*) and the 2D body part observations.

Here as well DNN have become the mainstream techniques, greatly improving over previous feature based methods (Chen et al., 2011b). This has led to the emergence of a large number of methods addressing 3D pose estimation from the monocular RGB case as well as in depth images. In the following we review relevant literature for this thesis grouping it by monocular RGB approaches and depth image based approaches. We also review literature for the multi-person settings. We focus on state-of-the-art estimating 3D body skeletons rather than approaches that estimate 3D mesh representations, e.g. (Xu et al., 2021).

2.3.1 Monocular 3D Pose Estimation

The monocular case problem is usually addressed as finding the 3D skeleton that when projected on the image plane, approximates the 2D detected landmarks. Estimating 3D points from 2D ones is an ill-conditioned problem: there are many 3D points configurations that can result in the same 2D projection. Therefore, methods have to rely on constraints related to physical measures and priors of plausible pose configurations.

Early approaches first localized body parts in the image and coupled these with 3D pose priors that model body part kinematics (Sigal et al., 2011; Ramakrishna et al., 2012). The problem is formulated as finding a set of body poses from an over complete dictionary **B**, such that their linear combination minimizes the projection error with respect to the 2D pose **x**

$$\min_{\alpha} ||\mathbf{x} - \mathbf{M}(\mathbf{B}\alpha + \boldsymbol{\mu}_y)|| + \lambda||\alpha||, \tag{2.2}$$

where $\boldsymbol{\mu}_y$ is the mean pose, **M** is the projection matrix and α is a vector of basis coefficients which have to be estimated. The pose dictionary **B** normally represents 3D pose basis computed from training data with PCA. Eq.(2.2) is normally solved by further imposing physical constraints such as anatomical proportions (Ramakrishna et al., 2012) and motion ranges (angles) of body's articulations (Akhter and Black, 2015).

Deep learning-based methods inspired from these early approaches relying in accurate CNN-based 2D pose estimation and NN regression schemes to solve the 3D pose estimation task. We group the literature into two main threads: similarity and learning-based methods.

Similarity-based Methods

Approaches in this category usually extend early works by still formulating the problem as a projection error minimization like Eq. 2.2, but use a deep CNN instead of classic machine learning predictors to accurately localize 2D body parts. Nevertheless, body part detectors are not perfect and still can introduce errors, i.e. missing body parts due to error or self-occlusion. Reasoning about occluded or missing body landmarks from 2D detections is problematic given that, at some degree, body anatomical properties are lost in the 2D projections.

One of the recent trends in the literature is to overcome missing landmarks by representing the prior information as pairs of 2D and 3D body skeletons. These methods rely in a large database of pairs of 3D skeletons with corresponding 2D projections in different viewpoints. See Figure 2.7 for an illustration. 2D and 3D skeletons are projected to a learned joint embedding where 2D skeletons (even incomplete ones) map to their complete 3D configurations (Li et al., 2015). Then, 3D pose inference from an incomplete 2D skeleton is reduced to compute a similarity search over the dataset of pairs using the learned embeddings. For example, the works presented in (Chen and Ramanan, 2017; Rogez et al., 2019a) follow this approach and use the *Nearest Neighbour* algorithm to efficiently perform the search.

Learning-based Methods

Methods in this thread leverage DNN models to directly predict the 3D locations of the body parts using a regression setting. Several type of approaches are in the literature that regress volumetric confidences (Pavlakos et al., 2017) or 3D coordinates (Habibie et al., 2019). Although simple, the 3D coordinate regression has proved to be an effective and efficient solution. We target the rest of this short review to methods predicting 3D coordinates of body parts.

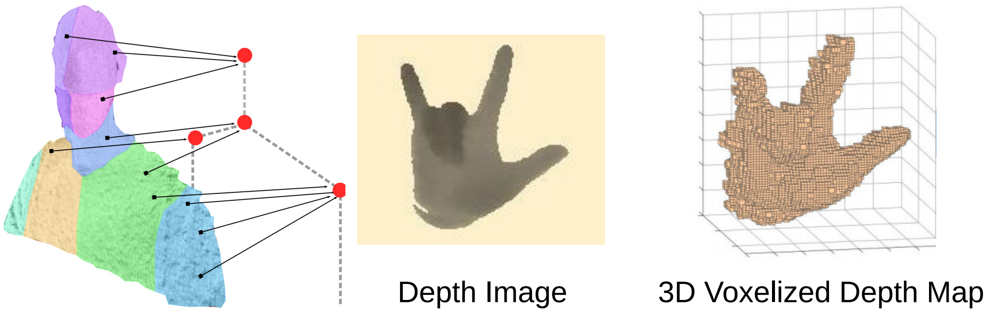
Regression methods receive as inputs any image observations, e.g. feature maps, image crops, etc. Perhaps the most simple approach is to use 2D body detections to regress 3D body parts with fully connected networks (Martinez et al., 2017b). The processing pipeline is simple: a neural network regressor f receives an input 2D body pose $\mathbf{x}_{2D}$ processes the input with a sequence of layers and computes the corresponding 3D pose.

In practice, the neural network regressor f can be composed of several stacked fully connected layers with *ReLU* activations and *dropout* regularization. The final fully connected layer (output layer) architecture can incorporate prior information about the structured dependencies of the human body which are convenient to reconstruct the 3D pose (Aksan et al., 2019). For example, the output layer can be divided into several small ones and specialize each one to predict an specific body part type. Then, to infer a body part type l_i, e.g. wrist, its corresponding network is fed with the prediction of parts l_j that are adjacent to it such as elbow and shoulder, following a skeleton model like in Figure 2.1.

More information about the body structure can be incorporated in the form of regularization during training. For example, the work presented in (Sun et al., 2017) introduces a composition loss that incorporates consistency on the anthropocentric dimensions. The loss penalizes kinematic chains (entire extremities comprising different body parts e.g. shoulder, arm and forearm), which lengths do not comply with the ones observed in the ground truth.

2.3.2 3D Pose from Depth Images

The depth data modality has also been largely exploited in the literature. Contrary to the monocular RGB case, the depth measurements removes the depth projection ambiguities and provides the 3D information of the scene via a point cloud, i.e. a list of X, Y, Z coordinates computed with the depth camera parameters. We review the most relevant classical and new deep learning-based literature.



(a) (b)

Figure 2.8 – (a) Offset regression voting for 3D pose prediction from depth surfaces. (b) Illustration of voxelized representation of a hand depth surface as computed in (Moon et al., 2018).

Early Classical Approaches

As with the monocular RGB color case, early techniques relied on optimizing an energy function to fit a 3D skeleton to image observations. Such methods search one-to-one correspondences between a point cloud and a general 3D mesh model of the human body, applying if necessary a series of transformations (rotation and translation) to the 3D mesh. One of the most popular search algorithms is the *Iterative Closest Point* (ICP) and has been widely used in 3D pose estimation (Ye et al., 2011).

Besides searching, one-to-one correspondences and transformations can also be learned from data. This setting requires a dataset of depth images paired with aligned 3D meshes. The strategy consist on learning a joint distribution of depth surfaces and correspondent vertices of the 3D model in an embedded space. At test time, instead of search, one-to-one correspondances are predicted, e.g. with random forest (Taylor et al., 2012). Learning correspondences was also the approach used by (Shotton et al., 2012). However, instead of predicting transformations, they follow an offset regression approach to localize body parts. First they classify pixels as a body parts. Then, 3D offsets are predicted from each surface pixel to the interior location of the body part skeleton. Weighted voting is used to generate the final 3D predictions of the skeleton. See Figure 2.8(a).

Deep Learning-based Approaches

Deep learning methods have largely followed the approach of predicting 3D coordinates from an input image crop. Deep network models are presented with either raw depth data (Guo et al., 2017) or a relatively expensive voxelized representation (Figure 2.8(b)). The goal is to compute an image representation that encodes 3D information from the depth image and exploit it to localize body landmarks. In theory, the representation can model global 3D information of the scene as well as rigid transformations. The work presented in (Haque et al., 2016) aims at learning a depth image representation that is invariant to different camera perspectives. Image patches are transformed with a CNN to a voxelized space where body parts from different camera views have the same semantic meaning. 3D coordinates of body landmarks are predicted from this voxelized space using a Recurrent Neural Network (*RNN*) that refines the 3D coordinates sequentially. 3D-specialized deep learning architectures have also been used. Another example is the work presented in (Moon et al., 2018) where depth images are transformed into a voxelized representation and processed with a 3D-CNN to predict the body landmarks in the form of 3D Gaussian likelihoods. Though the voxelization of the 3D space seems an appealing 3D data representation, most of the space tends to be empty. As a consequence, the 3D-CNN is a rather computational expensive approach.

2.3.3 Multi-person 3D Pose Estimation

Multi-person 3D pose estimation can leverage from any of the previous methods by taking one single person instance at a time to predict 3D coordinates. With the advances on multi-person 2D pose estimation and efficient object detection architectures like the Fast Regional CNN (R-CNN), a new trend has emerged where multi-person 3D pose estimation is addressed in a single pass. These methods heavily rely in the anchor proposal scheme of the R-CNN to identify person region proposals and further predict 3D pose. Usually, the pipeline consist on first generating anchor proposals of image regions with individuals. Then, a filtering process is applied to discard ambiguous anchors where people appear heavily overlapped. Finally, a 3D coordinate regression network is used to generate the body landmark coordinates for each persistent anchor using their corresponding image features. Techniques of this kind usually exploit the filtering process to enhance the 3D predictions. The work in (Rogez et al., 2019b) filters proposals by classifying the regions as potentially containing an "anchor pose" from a dictionary of poses. The identified anchor poses are further refined with a neural network regressor to output the predicted 3D pose. A fine-level strategy is adopted by (Benzine et al., 2020) using a 2D pose-aware filtering process. Instead of relying in entire regions to measure overlap, their model is trained to disregard region proposals with heavily overlapped 2D body landmarks.

2.4 3D Motion Prediction

An important component for human behaviour understanding resides in the ability of a system to comprehend the human motion. Human motion prediction is an essential part of the human behaviour understanding pipeline given that it provides the capacity to estimate human intentions. Despite its applications in different domains, e.g. visual surveillance or human-robot interaction, human motion prediction remains a challenging task. On one side, human motion is a non-linear system with high degree of uncertainty due to the articulated structure of the body. Although the movement of the body parts is highly correlated, these properties are hard to model in learning systems. On the other side, body movement is largely correlated with the activity being performed and influences person intentions.

We understand as human motion a continuous sequence of time dependent observations of a person performing a certain activity, i.e. walking, taking a cup, etc. Different works have studied motion prediction from temporal sequences of different kinds of observations, i.e. 2D body landmarks (Raaj et al., 2019), and 3D body landmarks (Martinez et al., 2017a). In this short review, we focus on motion prediction from sequences of 3D body landmarks such as in the setting illustrated in Figure 2.9.

A motion sequence is considered as the temporal sequence of 3D poses $\mathbf{X} = \{\mathbf{x}_{1:T}\}$ where $\mathbf{x}_t \in \mathbb{R}^N$ are N-dimensional pose vectors. Generally, the sequence has been modelled in an autoregressive setting factorizing the joint probability as a product of conditionals

$$p(\mathbf{X}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{1:t-1}). \tag{2.3}$$

At time step t the next pose is predicted given all the past poses. Early classical machine learning methods made Markovian assumptions to ease inference by writing the temporal conditionals as $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ (Lehrmann et al., 2014; Fleet, 2011). Additionally, the dynamics of the body assumed smooth transition between poses of two consecutive time steps such as $\mathbf{x}_{t+1} = \Psi(\mathbf{x}_t) + \eta$, where η is a white noise random variable and $\Psi(\cdot)$ is a transition function. The transition function $\Psi(\cdot)$ has adopted different models from linear functions (Fleet, 2011) to Gaussian processes for regression (Urtasun et al., 2006). For example, the work presented in (Sigal et al., 2011) coupled 3D pose estimation and motion by modelling the relationships between body parts with a mixture of Gaussian kernels that accounted for body kinematic constraints. Belief propagation was used during inference to estimate motion and pose.

Amid deep learning, human motion prediction has been addressed with deep learning-based methods such as *Recurrent Neural Networks* (RNN) with LSTM or GRU units. We focus the rest of this review in more recent deep learning-based methods.

2.4.1 Deep Learning for Motion Prediction

Recently, a family of methods using deep RNN have shown great improvements in human motion prediction. The biggest advantage of these approaches is that they make less assumptions about the dynamics of the human body and motion, and instead they attempt to learn these from data. The current available datasets with millions of examples (Ionescu et al., 2014; Mahmood et al., 2019) have allowed learning deep models with outstanding prediction performance for short and long term time horizons.

Deep learning-based methods address human motion prediction as a sequence-to-sequence modelling problem with a encoder-decoder architecture where the goal is to model the conditional probabilities $P(\mathbf{Y}|\mathbf{X};\theta)$ with model parameters θ. We can set the problem as follows:

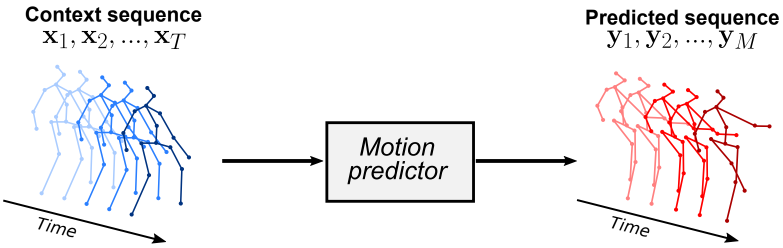
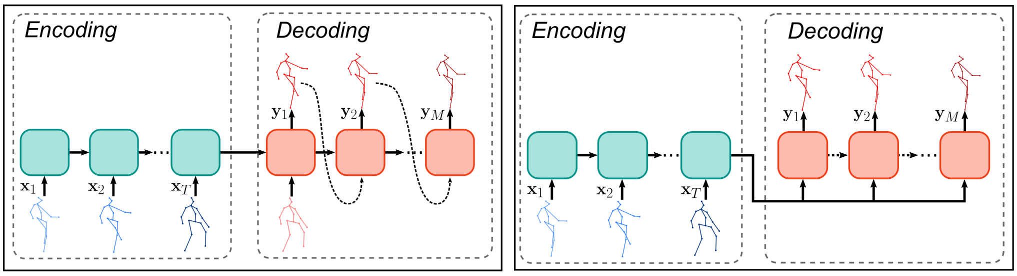

Context sequence
$\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$
Time
Motion predictor
Predicted sequence
$\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M$
Time

Figure 2.9 – Simplified overview of human 3D motion prediction. A temporal sequence of T continuous 3D pose elements are input to a motion predictor. The predictor generates the most likely immediate sequence of M 3D pose elements.



(a) (b)

Figure 2.10 – Illustration of (a) autoregressive and (b) non-autoregressive motion prediction.

given a temporal sequence of body poses $\mathbf{X} = \{\mathbf{x}_{1:T}\}$ we seek to predict the most likely immediate following motion sequence $\mathbf{Y} = \{\mathbf{y}_{1:M}\}$, where $\mathbf{x}_t, \mathbf{y}_t \in \mathbb{R}^N$ are N-dimensional pose vectors. We group the different techniques in autoregressive and non-autoregressive methods.

Deep Autoregressive Methods

Figure 2.10(a) illustrates the autoregressive setting with an encoder-decoder architecture. First, the encoder takes the conditioning motion sequence $\mathbf{x}_{1:T}$ and computes a contextual representation $\mathbf{z}_{1:T}$. The decoder then generates pose vectors $\mathbf{y}_t$ one by one taking the context $\mathbf{z}_{1:T}$ and its previous generated pose vectors $\mathbf{y}_{\tau<t}$.

Early deep learning approaches used stacks of RNN units to form encoder and decoder architectures. Though they achieve good overall performance, predictions suffered of a catastrophic drift. The drift refers to the convergence of pose predictions to the mean pose in long term time horizons as a cause of the error accumulation by predicting in an autoregressive fashion. The work in (Fragkiadaki et al., 2015) addressed the drift by including an error scheduling to gradually add Gaussian noise to the input observations and increase robustness to noisy observations. This scheduling however, turns out very hard to tune in practice. Other methods include an adversarial loss to reduce error accumulation and enhance the quality of predictions at long term (Gui et al., 2018; Hernandez et al., 2019). For example the work in (Hernandez et al., 2019) proposes a *Generative Adversarial Network* (GAN) as an approach to generate plausible motion at longer horizons. Their architecture consist on a convolutional generator designed to preserve temporal coherence and a set discriminators that enforce anthropomorphism and realistic motion in longer horizons. Instead of training the GAN to perfectly match the ground truth, the framework is trained to match the ground truth distribution in a human motion manifold. While these adversarial settings boost performance, their training is rather complicated and hard to stabilize due to the use of multiple discriminators.

Besides long term error accumulation, short term predictions also suffer from large discontinuities between last observation $\mathbf{x}_T$ and first prediction $\mathbf{y}_1$. The discontinuity had been shown to be so severe that simple running average and zero-velocity (predicting always $\mathbf{x}_T$) baselines performed better than complex RNN models. This shortcoming has been addressed by modelling the velocities between predictions (Martinez et al., 2017a) or introducing the body structure dependencies during decoding (Aksan et al., 2019). The method introduced by (Martinez et al., 2017a) addressed the task with a single layer RNN architecture shared between encoder and decoder. To reduce the discontinuity they include a residual connection in the autoregressive decoding between predictions $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$. The residual connection has the effect of incorporating velocities and adds smoothness between consecutive predictions.

Since the breakthrough of the transformer neural network (Vaswani et al., 2017), attention-based approaches have recently gained interest for modelling human motion. Given their infinite memory, attention modules are capable to identify individual parts of the complete input sequence that are relevant for prediction. For example, the work presented in (Wei et al., 2020) exploits a self-attention module to attend the input sequence with a sliding window containing a small subsequence of the same input. Ideally, attention should be larger in elements of the input sequence that repeat with time, even for not periodic activities. Prediction works in an autoregressive fashion using a GCN performing from the attention embeddings. Along the same line (Aksan et al., 2021) introduces an spatio-temporal self-attention module to explicitly model the spatial components of the sequence, in an attempt to model the relationships between the different body parts. The approach processes the sequence with two separate self-attention modules: a spatial and a temporal module, working over a sequence of 3D points and over a sequence of 3D skeletons respectively. The resulting attention maps are aggregated with feedforward networks and elements are predicted in an autoregressive manner.

Non-autoregressive Modelling

Most neural network-based models for sequence-to-sequence modelling use autoregressive decoding: generating entries in the sequence one at a time conditioned on previous predicted elements. Such approach have two major shortcomings. First, autoregressive models are prone to propagate prediction errors. Elements in the predicted sequence are conditioned to already spurious observations, which naturally increment their error contents. Second, autoregressive modelling is not parallelizable causing deep learning models to be more computationally intensive. An example of this shortcoming is the transformer neural network used in recent machine translation (Vaswani et al., 2017; Radford et al., 2019). During training, the architecture allows parallelization with *look ahead* masking. Yet, at testing time, the use of an autoregressive setting makes it difficult to leverage the parallelization capabilities. Recent efforts have sought to parallelize decoding with transformers to allow for an efficient testing phase in different domains such as in machine translation (Gu et al., 2018) and in visual object detection (Carion et al., 2020).

In principle, non-autoregressive decoding in a encoder-decoder architecture functions as illustrated in Figure 2.10(b). First, the encoder generates a contextual representation $\mathbf{z}_{1:T}$ of the input sequence. Then the decoder uses it to generate M predictions in parallel dropping the autoregressive connectivity. This approach exhibits complete temporal conditional independence as prediction only depend on the source sequence. Hence, it is a poor approximation to the true target temporal distribution. To overcome this limitation recent efforts in machine translation include temporal information by modelling the ordering of the target sequence (Bao et al., 2020) or by predicting *fertilities* to replicate elements from the input sequence (Gu et al., 2018).

Parallel decoding has also being explored in the human motion prediction domain. Clearly, the most challenging aspect is to represent the temporal dependencies for decoding predictions. Most of the solutions in the literature provide additional information to the decoder that account for the temporal correlations in the target sequence. Different methods have been proposed relying in decoder architectures that exploit temporal convolutions (Li et al., 2018), feeding the decoder with learnable embeddings (Li et al., 2021), or relying in a representation of the sequence in the frequency domain (Wei et al., 2019). The work presented in (Wei et al., 2019) represents the temporal dependencies using the Discrete Cosine Transform (DCT) of the sequence. During inference a *Graph Convolutional Network* (GCN) predicts the DCT coefficients of the target sequence. However, to account for smoothness, during training, the GCN is trained to predict both input and target sequence DCT coefficients. The approach presented in (Li et al., 2018) performs a similar approach by modelling separately short term and long term dependencies with temporal convolutions. Their decoder is composed of a short term and long term temporal encoders that move in a sliding window. Short and long term information are then processed by an spatial decoder to produce pose frames.

2.5 Domain Adaptation

When performing supervised learning we commonly assume that samples used for both training and testing phases follow the same probability distribution. However, this assumption is not always fulfilled in real world applications due to the difficulty in collecting data with annotations in the target scenario (*target domain*). One solution consist of using available data in one or more related *source domains*, learn a predictor in these domains and apply it to the target domain. However, such an approach provokes a degradation in the generalization of the trained model since our training and testing data distribution assumption is not fulfilled.

This situation gives rise to the situation called *covariate shift*(Sugiyama and Kawanabe, 2012).

The rest of this section is organized as follows. First, we review state-of-the-art methods that address the need of data with annotations by using synthetic data, which gives rise to the covariate shift. Then, we introduce a family of approaches that apply deep learning-based domain adaptation techniques to address the mismatch between training and testing distributions.

2.5.1 Learning from Synthetic Data

Learning models for human pose estimation requires data with high quality body joint annotations for supervised learning. For recent deep learning approaches that aim to learn powerful deep networks, having a large and varied dataset with annotations is even more vital. Despite the available datasets, some applications might need a different set of input data or annotations, e.g. 3D landmarks, different camera perspective, body segments, etc., making collecting annotated data very expensive. A trend in human pose estimation research overcome this challenge with the use of computer graphics to synthesize training data. This allowed to improve the state-of-the-art in 3D human pose estimation in the wild (Chen et al., 2016) and to target specific surveillance scenarios like security airlocks (Villamizar et al., 2018, 2020)

The grounds of data synthesis with computer graphics is to use motion simulation with a set of 3D human characters. In most cases, this implies to use a motion re-targeting algorithm to transfer motion sequences to the 3D characters. The motion sequences have been previously recorded with Motion Capture (MoCap) sensors inside a studio. Images are then generated by rendering graphics scene from a set of virtual cameras. Given that all information about the geometry of the scene can be accessed, annotations (2D and 3D landmarks, 3D meshes, segmentation masks, etc.), can be extracted at rendering time. However, achieving a level of image realism is challenging and requires expert knowledge about the final target scenario, e.g. statistics of depth sensing errors (Shotton et al., 2012). Some works have address this challenge by including indoor type of objects in the scene (Varol et al., 2017) or using realistic gaming engines (Fabbri et al., 2018).

2.5.2 Unsupervised Domain Adaptation

Domain Adaptation (DA) refers to *transfer learning* aiming at learning a predictor in the presence of the covariate shift. In general, it is assumed that the task in the source and target domains is the same and that the domains are related but not identical. In computer vision applications the situation arises for instance from changes in lighting conditions, background, and object pose but the mismatch can be more severe when, for example, image types are different (color and depth) or target sensors are subject to different types of noise (Patricia et al., 2017). An example more related to this thesis is when the covariate shift emerges when learning models from synthetic images and test them with real images.

The literature in domain adaptation can be separated depending on whether the domains are *homogenous* and *heterogeneous*. The former case emerges when source an target are represented in the same feature space, while the latter appears when domains can have different representations. It can also be distinguished depending on the availability of domain labels, leading to unsupervised or semi-supervised methods.

The rest of this section summarizes techniques of domain adaptation for recent deep learning applications. We focus in unsupervised domain adaptation which assumes that no labeled data is available in the target domain.

Unsupervised Deep Domain Adaptation

The premise of a domain adaptation technique is often to learn a data representation that is invariant between domains. Unsupervised domain adaptation pursues to learn a model in the target domain for which there is data with no labels, using the available labeled data in the source domain to achieve the prediction task. One of the pioneer works in unsupervised domain adaptation is the work presented in (Glorot et al., 2011) applied to sentiment classification. It introduced an approach that learns a feature representation of the source and target domain by reconstructing the input data with denoising auto-encoders. The learned auto-encoder representation is then used to classify sentiments from reviews with SVM.

Current deep learning methods usually follow a DNN architecture with two branches: a prediction and an adaptation branch, both sharing a feature extractor network. The prediction branch is trained for the task at hand, e.g. classification. The adaptation branch is trained to encourage a common feature space for both domains. Additionally, it is in charge of measuring the difference between the feature representation of both domains, evaluating the alignment of data transformations by a discriminative process.

The adaptation component most broadly used is an *adversarial discriminative model*. This approach learns a common feature space between domains by aiming to confuse a domain discriminator (Long et al., 2016; Ganin et al., 2016). For example, the work in (Ganin et al., 2016) proposes a discriminative approach to learn domain invariant features for image classification. The method jointly learns an object class and domain predictors relying on shared features between the two tasks computed with a deep CNN. Learning is performed by minimizing the image classification loss while maximizing the error on domain classification loss. The learned feature representation should encode global image information that is invariant across input domains.

In addition to achieving invariance representation at the global image level, other approaches seek fine grained invariance at local pixel level (Hoffman et al., 2018) or at object class level (Venkateswara et al., 2017). The work presented in (Hoffman et al., 2018) employs an image-to-image translation approach to enforce consistency in the semantic feature space during adaptation. The goal is to reconstruct the input image from the adapted feature representation, hence enforcing that detailed local pixel statistics are taking into account during adaptation.

2.5.3 Beyond image classification

The majority of domain adaptation methods in computer vision focus on image recognition from color images, with relatively few working in another modality or tasks. Moreover, source and target domains differ mainly in objects perspective, image background and lighting. The work in (Venkateswara et al., 2017) goes a step further by transferring domain knowledge from RGB images to sketches.

Very few works explore domain adaptation in other modalities. The work in (Patricia et al., 2017) studies state-of-the-art domain adaptation techniques for object classification from depth images. It uses different RGB-D datasets and attempts to transfer domain knowledge for classification between modalities. However, it shows that there is an intrinsic difficulty in performing adaptation given that noise in depth images is significantly more persistent and different between sensors compared to that among RGB images.

Amongst the non-classification task related to this thesis is the approach of (Chen et al., 2016) which applies domain adaptation for 3D human pose estimation learned from synthetic data. The method follows the domain adversarial fashion with a two branched architecture, i.e. a 3D pose regressor and a domain classifier sharing a feature extractor.

2.6 Model Compression and Knowledge Distillation

Recent advances in computer vision tasks rely in increasingly deep CNN architectures in order to achieve better performance. In the ImageNet challenge (Russakovsky et al., 2015), where the task is image classification, in each competition a new deeper and larger DNN architecture was introduced beating previous challenge results. The compositional advantage of deep learning methods allows to design very deep models, with some of them comprising hundreds of layers (He et al., 2016). Figure 2.11 shows the classification accuracy of the most popular models applied to the ImageNet challenge, and compares them in terms of floating operations and number of parameters.

Although deep networks produce outstanding results, these design choices give rise to certain disadvantages. We note that DNNs with large model capacity and many parameters achieve good generalization performance. However, larger architectures imply more need for data and engineering to setup the training hyper parameters, e.g. learning rate, regularization, etc. Additionally, these over-parameterized models have no incentive for speeding up inference. Thus, the models are binded with large computational costs, hindering their practical application in scenarios with very low computational budget as in HRI.

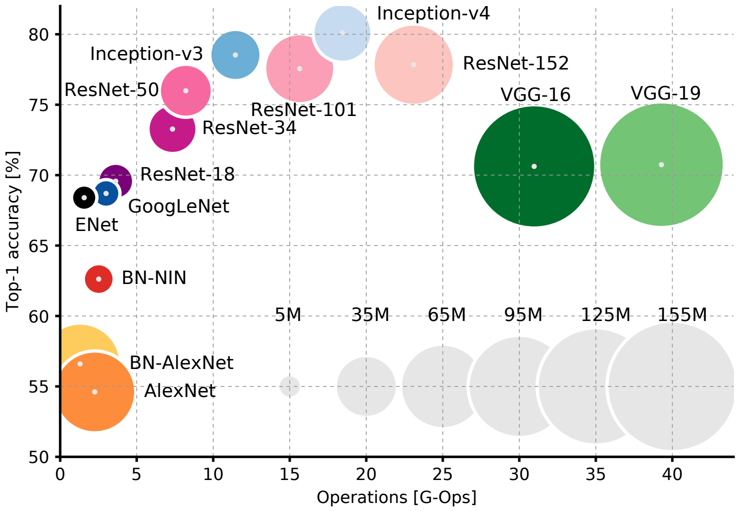


Figure 2.11 – Accuracy vs size comparison of state-of-the-art architectures in ImageNet challenge for image recognition (source (Canziani et al., 2016)).

In this section we review a set of methods that aim to overcome the large computational cost of DNN models by reducing their size without necessarily degrading performance. We group the methods in model compression and knowledge distillation.

2.6.1 Model Compression

Model compression aims at removing unnecessary parameters in a DNN model to produce a lighter and more efficient version. The compression process can be applied in different ways. On one hand, it can be applied to already trained models by removing some of its parameters that are found as unnecessary. On the other hand, lightweight architectures designs aim at directly increasing speed by exploiting different convolution strategies.

Channel pruning methods are among the most used techniques for model compression of pre-trained models. Channel pruning is performed during training and aims at removing convolutional filters that produce statistically very low activations during the learning process (Molchanov et al., 2017).

Other recent lightweight architectures have been proposed in the field of image classification aiming at reducing computational cost without hurting the overall performance. For example, the MobileNet (Howard et al., 2017) architecture factorizes standard convolutions in order to reduce the model size and increment the speed. For a kernel of size D_K and feature map resolution of $D_F \times D_F$ the standard convolution have a cost of $D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$, for M and N input and output channels. The MobileNet architecture factorizes the convolution into depthwise and pointwise convolutions, which perform per channel convolution and merging separately. The final reduction of computation is of $1/N + 1/D_K^2$. Similarly, the SqueezeNet model (Iandola et al., 2016) constrains the number of input feature channels of layers with large kernel size. These architecture designs have made a tremendous gain towards efficiency, with some of them being able to run 50 times faster than their deeper counterpart with small loss in performance.

2.6.2 Knowledge Distillation

Knowledge distillation is a line of research that pursues reducing the size of models without hurting performance. The objective of knowledge distillation is to transfer the generalization ability of a large and accurate DNN model, named the *teacher*, to a small and lightweight model, called *student*. The main idea is that this generalization ability the teacher lies in the activations of the model rather than in its parameters and therefore such activations contain valuable information about how the model tends to generalize.

The pioneering work of (Hinton et al., 2015) showed how the student model can acquire the teacher knowledge using the teacher activations as soft targets. These activations implicitly contain information about the difficulty of the samples, e.g. ambiguity between classes, uncertainty due to difficult imaging conditions. The learning framework consists of mimicking the teacher activations by minimizing a weighted average of losses based on the ground-truth labels and the prediction of the teacher model.

Other forms of distillation focus on using the teacher activations to guide the learning stage of the student. For example, matching the teacher and student Jacobians of the learning objective as in (Srinivas and Fleuret, 2018) to regularize distillation, or using the teacher activations as bounds for distillation in regression settings (Chen et al., 2017). Recent works have also pursued mimicking inner teacher activations in an attempt to also distill data representations. For example, the work in (Romero et al., 2015) have proposed to introduce "hints" as an attempt to additionally mimic the activations of a given hidden layer of the teacher.

2.7 Conclusion

This chapter presented the state-of-the-art of methods for the different components of the human behaviour pipeline that we study in this thesis.

First, we introduced the task of 2D pose estimation. Current deep learning-based techniques rely in deep and large CNN architectures trained with large amounts of color images with annotations. This practice however prevents to deploy CNN-based systems in applications with low computational budget like HRI scenarios. In contrast, we focus on 2D pose estimation using depth images with lightweight efficient CNN architectures. We design our architectures with a cascade of detectors and heavily use Residual, Mobilenet and Squeezenet modules allowing to perform fast inference in real-time. A novelty of our work is that we apply knowledge distillation for body landmark detection by strongly coupling distillation with the process of refining predictions, compared to simple knowledge distillation or learning with hints. This process allows us to go even further in efficiency as we reduce computational cost without hurting performance.

We presented different approaches that use synthetic images to address the lack of data with annotations for training. In our work we also generate synthetic data for our learning step and we investigate domain adaptation techniques to address the mismatch between training and testing distributions and prevent our learned models from the performance drop. However, different to most domain adaptation works, which focus in object classification tasks in settings with few domain differences (objects perspective, image background and lighting), we analyse unsupervised domain adaptation for a regression task. Additionally, we contrast its limitations with simple finetuning and data transformations approaches.

We discussed different deep learning-based trends to predict human pose in 3D coordinates. One of the most popular approaches is to make predictions using a regression scheme from 2D points or image crops. A drawback of these methods is that they predict the 3D coordinates with respect to a root joint that is assumed to be known in advance, or which in practice needs to be predicted as well. Moreover, to handle the multi-person case, a person detector is still needed as a first step followed by forward passes of the 3D pose predictor for each person instance. In the case of using image crops as inputs the methods rely on preprocessing steps such as voxelization which can be relatively heavy, increasing the computational cost. In our work, we perform 3D pose estimation from depth images exploiting the depth information to decouple the task in two main steps:2D multi-person pose estimation and 3D pose regression. Our 2D pose estimation approach benefits from accurate and efficient CNN architectures while 3D pose estimation is performed efficiently by directly regressing the 3D coordinates from the 2D ones in two substeps: 1) a simple but effective scheme which lifts the 2D estimates to 3D using the depth image information and pose priors(to handle partial occlusion); and 2) a novel efficient residual pose 3D regression method that works on this set of points. Our approach is computationally lighter for multi-person HRI settings since compared to CNNs applied to image crops for 3D pose prediction the cost of our 3D regression scheme is much smaller and proportional to the number of people in the scene.

Finally, we reviewed a family of deep learning-based approaches for human 3d motion prediction in both settings, autoregressive and non-autoregressive decoding. We noted that most methods rely in autoregressive settings using RNNs. However, these methods suffer from error accumulation arising by predicting motion sequences from spurious predictions. Additionally, autoregressive decoding prevents from parallelizing inference and increases the computational cost of models such as transformers. Our work on 3D motion prediction lies within the techniques that use attention for prediction. However, our approach is framed in a non-autoregressive decoding, using a transformer with self- and encoder-decoder attention. In contrast, to methods using only self-attention modules, this allows us to exploit the infinite encoding memory of transformers to find elements in the input that are relevant for prediction while allowing for parallel prediction to reduce computation time during inference.

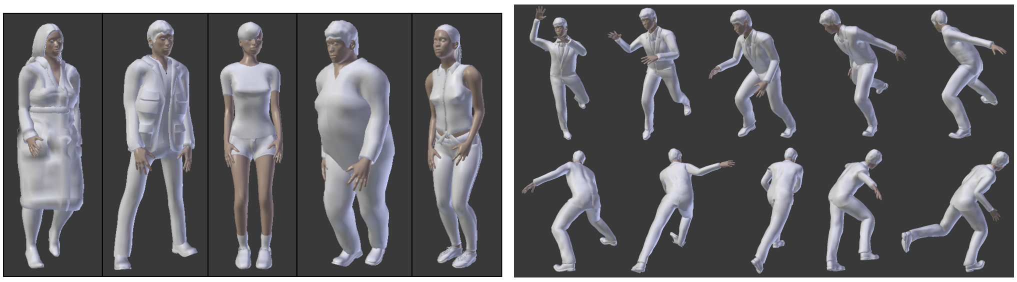
Contrary to state-of-the-art methods with non-autoregressive decoding we not incorporate any prior information of the temporality of the sequences, like using the DCT coefficients of the sequence, and let the transformer to learn other suitable representations.

3 Depth Image Databases

A critical element for any machine learning-based method is to have a varied and representative dataset for the learning stage. In the human pose estimation and behaviour analysis literature earlier datasets contained simple single-person image centered examples with very few images for training and testing, e.g. the LSP dataset (Johnson and Everingham, 2010) with only 1K. In recent years an increasingly number of datasets with annotations have been made available for supervised learning. Images in such datasets are labeled with the location of body parts in image coordinates, instance segmentation mask, and activities, and are largely more varied exhibiting more challenging setups like multi-person scenarios, profile views, occlusions, etc. Some common datasets contain thousands or even millions of images collected from the internet (Lin et al., 2014; Andriluka et al., 2014) or obtained with Motion Capture (mocap) environments (Ionescu et al., 2014).

Although currently most databases use color information (RGB), in this thesis, we focus on depth images. Contrary to RGB images, which have a high diversity of color and texture content, a depth image contains mainly shape information making them relatively simpler while still containing rich and sufficient information for efficient human pose estimation, as shown by Shotton et al. (Shotton et al., 2012) They may thus necessitate less complex models to accomplish the pose estimation task.

This chapter introduces the datasets we use throughout our experiments. We consider two types: a synthetic depth image dataset which we generate with the aim of addressing the need of a large training corpus, and real image datasets that we use for finetuning and evaluation. First, we describe our process to generate our large scale dataset of synthetic depth images. Then, we describe the real depth image datasets.



(a) (b)

Figure 3.1 – (a) Sample 3D characters with different body features and outfits (b) 3D characters retargeted with different CMU-Mocap sequences.

3.1 Synthetic Depth Image Dataset

3.1.1 Synthesizing Images of People

In spite of the existing datasets, large quantities of labeled data are not always available for specific application or are expensive to produce. Therefore, researchers have investigated the use of computer graphics to synthesize large datasets when the application requires a different set of annotations, e.g. human 3D pose estimation (Fabbri et al., 2018; Chen et al., 2016) or 3D character manipulation (Lassner et al., 2017). Addressing the lack of data with synthetic images brings the following benefits: high quality annotations are extracted at rendering time, and variability in the dataset (i.e. human shapes, body pose, viewpoints, etc) can be introduced to match the target scenario.

In this thesis, we address the lack of training data for human pose estimation from depth images by synthesizing a large scale dataset of depth images of people with high quality body landmark annotations. The generated data forms part of the DIH dataset which we released to the public for research purposes [1]. The following sections describe the randomized rendering pipeline approach (Martínez-González et al., 2020a) that we proposed to to synthesize our data.

3.1.2 Randomized Rendering Pipeline

The challenge in generating synthetic images for model learning is how to automatically introduce enough variability and to achieve a certain level of image realism. For the human pose estimation setting, we need to consider the following points:

- The human body exhibit a large range of pose configurations.
- Physical features of people and clothing increase the variability of body shape.

---

[1] https://www.idiap.ch/dataset/dih

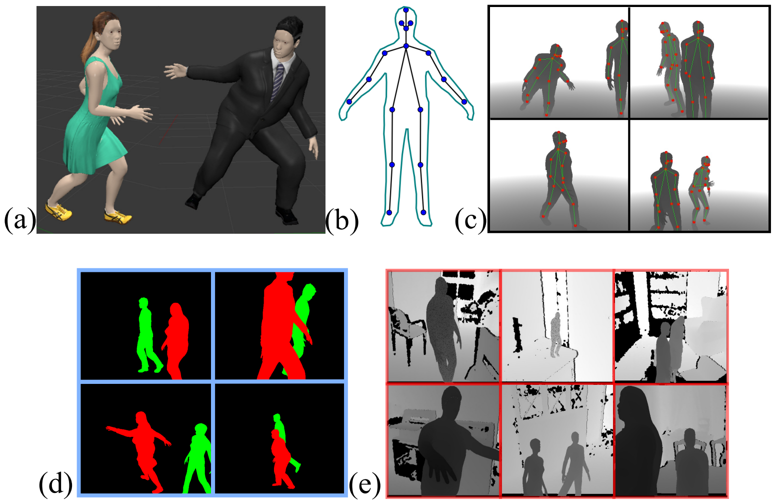


Figure 3.2 – Randomized image synthesis pipeline. (a) Sample 3D characters with different poses and outfits; (b) skeleton model; (c) rendered synthetic depth image sample; (d) examples of generated colored depth mask for synthetic images with more than one person; (e) examples of training images, combining synthetic generated bodies with real background images.

		Height			Body Mass		
Gender	QTY	Short	Average	Tall	Skinny	Average	Heavy
Men	12	1.60 m	1.75 m	1.90 m	weight 70%, muscle 20%	weight 100%, muscle 60%	weight 150%, muscle 30%
Women	12	1.55 m	1.65 m	1.75 m	weight 70%, muscle 20%	weight 100%, muscle 60%	weight 150%, muscle 20%

Table 3.1 – Summary of the physical features contained in the 3D characters collection. The models were created with the Makehuman (Makehuman Online Comunity, 2018) software. The physical features of the 3D models comprise different heights and different body weight.

- People in images may appear from different camera view perspectives.

- Real world scenarios normally exhibit multiple people in the images.

We address these points relying on motion simulation. Our randomized rendering pipeline is implemented using the computer graphics software Blender (Blender Online Community, 2017). In short, we perform motion re-targeting of motion capture sequences to a collection of 3D human characters. Then, we generate depth images from different camera perspectives by rendering the camera depth buffer. In the following, we provide a more detailed explenation of each of the components of our randomized pipeline.

Variability in Body Shapes

We created a collation of 24 3D human characters created with the Makehuman (Makehuman Online Comunity, 2018) design software. These characters show variation in gender, heights and weights, and were dressed with different clothing outfits to increase shape variations (skirts, coats, pullovers, etc.). Table 3.1 provides a summary of the physical features of the different 3D characters. See Figure 3.1(a) for some examples.

Variability in Body Poses

Motion simulation is used to add variability and to expand the range of feasible body pose configurations. We performed motion re-targeting from motion capture sequences taken from the CMU Mocap dataset (CMU Motion Lab, 2007). Among the available actions, we selected the following ones as most representative of our target scenario: walking, jogging, jumping, stretching, turning and various sport gesturing (shooting, blocking, etc). See Figure 3.1(b) for some examples of 3D characters with re-targeted motion.

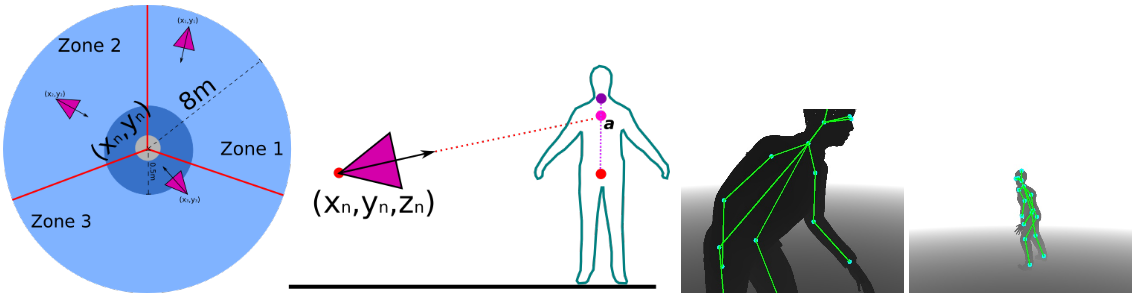
Variability In Viewpoint

Virtual cameras are randomly positioned in our virtual scenario to increase the variability of the dataset. The approach is summarized in Figure 3.3(a). We proceed as follows: first, the 3D character is placed at a fixed reference point $p = (x_r, y_r, z_r)$, and we define a recording zone (where we can set cameras) as a circle of $8m$ radius, centered at the character reference point. We split the zone in three and randomly place a camera in each considering a margin between the character and the camera. The coordinates of the cameras are randomly chosen to be inside the recording zones in the range $0.5m \leq \sqrt{x_c^2 + y_c^2} \leq 8m$ and height $z_n \in [1.3, 1.8]$. Finally, the camera's orientation is initialized such that it looks at a point a randomly chosen between the spine and the neck of the 3D character. The orientation of the camera does not change over the course of the rendering. Note that there is no calibration between the cameras, rather each generates independently images and annotations.

The outcome of this process provides us with images containing observations at different scales, partial observations of the body and different points of view that further increase the variability in our synthetic dataset. Some rendered examples are shown in Figure 3.3(b).

Synthesis with Two People Instances

In real world scenarios, specially in HRI ones, multiple people can be observed in the field of view of the camera sensor. In our dataset, multi-person pose estimation scenarios are covered by adding two 3D characters to the rendering scene. During synthesis, two models are randomly selected from the character database and placed randomly in the virtual scene, but keeping a minimum distance between them to avoid checking for collision. Figure 3.2(c)



(a) (b)

Figure 3.3 – (a) Synthesizer scenario configuration. The recording zone is divided in three zones at which a camera is randomly placed and randomly oriented towards the 3D character. (b) Rendered samples with our randomized pipeline contain variation in viewpoints, scale and partial observations.

shows some rendered examples.

Annotations

During rendering we extract the location of 17 body landmarks following the skeleton shown in Figure 3.2(b). The landmarks are *head, neck, shoulders, elbows, wrists, hips, knees, ankles, eyes*, and were extracted in 3D camera and 2D image coordinates.

We provide visibility labels for each of the body landmarks in the image. A body landmark is set to visible if it is inside the image bounds and by thresholding the distance between the landmark and the body surface point that projects onto the image at the same position than the landmark. We generate three semantic meanings for visibility labels: a) visible landmarks, b)invisible landmarks outside of the image bounds and, c) invisible occluded landmark. In our experiments a landmark is only treated as visible or invisible and such information is used to generate training ground truth.

We also provide person instance color labeled silhouette masks which can be used to determine keypoint visibility and to perform data transformations during training, e.g. adding pixel noise or fusing with real backgrounds (Section 3.1.4). See Figure 3.2(d) for some examples.

3.1.3 S-DIH Dataset

The generated synthetic dataset consists of 264,432 images of people performing different types of motion under different viewpoints with 71,711 images displaying two people. We divide the dataset into training, validation and testing folds. Table 3.2 shows the number of images per division fold. Some examples are shown in Figure 3.2(c).

Data fold	Training	Testing	Validation
S-DIH	230934	22333	11165
R-DIH	6338	3927	4828
R-DIH (annotated)	1750	1000	750

Table 3.2 – Summary of the number of images in the DIH dataset. S-DIH represents the synthetic part of the DIH dataset for which we have all the annotations. R-DIH is the real part of the DIH dataset. We have manually annotated only a subset of images (marked with annotated).

3.1.4 Real Background Fusion

Note that a realistic dataset needs as well realistic background content. A solution would be to generate random background content by randomly placing 3D objects in the rendering pipeline. This is however non-trivial as a large variety of objects is needed to cover all expected backgrounds. Instead we propose to use background images from the target depth sensors. This has two main advantages

1. A large variety of real background depth images (which do not require ground-truth) with different contents can be easily collected than generating synthetic body images;
2. The resulting data will already contain sensor specific information that will contribute in the generalization capabilities of the learned models.

As background images, we consider the dataset presented in (Silberman et al., 2012) containing 1367 real depth images recorded with a Kinect v1 and exhibiting depth indoor clutter, and we divided it into training, validation and test folds. When training our deep learning models, images were produced on the fly by randomly selecting one depth image background and body synthetic images, and generating a depth image using the character silhouette mask. The algorithm verified there was a sufficient depth margin between the body foreground (synthetic human silhouette) and the background, adding if necessary an adequate depth constant value to the entire background image. While crude, this approach resulted in more realistic data than the synthetic ones. Some examples are shown in Figure 3.2(e).

3.2 Real Depth Image Datasets

Depth imaging is usually the result of a triangulation process in which a series of laser beams are cast into the scene, captured by an infrared camera, and correlated with a reference pattern to produce disparity images and finally the distance to the sensor. As a result, the image quality greatly depends on the sensor specifications like measurement variance, missing data, surface discontinuities, etc.

Given that our aim is to perform pose estimation in real HRI scenarios we need datasets to adapt our models and evaluate their performance in real depth images. We considered three datasets. The first one corresponds to a set of video recordings with people interacting with a humanoid robot, i.e. a HRI scenario. The other two are known public benchmark datasets with single and multi-person settings. We describe each of these datasets in the following.

3.2.1 R-DIH Dataset

We consider realistic HRI scenarios of people interacting with a humanoid robotic platform (Pepper). To this end we performed a set of indoor recordings of people interacting with the robot and between them in a natural way. Our recording setting was composed of the robot depth camera (Asus Xtion), as well as two external sensors (Kinect v2 and Intel D435) as shown in Figure 3.4(b).

The recorded dataset contains 16 indoor sequences of up to three minutes composed of pairs of registered color and depth images. They display up to three people captured at different distance from the robot with different levels of occlusion and scene backgrounds. A total of 9 different participants were involved in natural HRI interaction situations (walking off and towards the robot, stretching hands and between person interactions). They wore different clothing accessories to add variability in body shape. Some examples of different scenarios are shown in Figure 3.4(a).

The recordings are divided into training, validation and testing folds comprising 7, 5 and 4 sequences respectively. We manually annotated a small subset of images for each fold: 1750, 750 and 1000 images within the training, validation, and testing folds, respectively. As annotations we provide 17 body landmarks in image coordinates and bounding boxes for each of the people in the images. Table 3.2 shows the number of images per fold.

Throughout our experimentation in this thesis we focus on the Kinect v2 sensor. Compared to other sensors like Intel D435 or Asus Xtion, it has a more accurate depth estimation and a larger field of view which is better for HRI analysis. The recorded dataset is part of the DIH dataset and is public for research purposes [2].

3.2.2 CMU Panoptic Dataset

The CMU-Panoptic dataset (Joo et al., 2017) is a large scale dataset for perception of social interactions of multiple people. It is composed of a multiview camera data recorded with RGB, Kinect 2 and audio sensors. The dataset provides RGB-D data with annotations such as 2D and 3D body joint locations, social activity annotations, cloud points and body joint trajectory as well as recordings metadata, e.g. transformations between cameras, global coordinates etc.

Among the many different provided recordings we focused in the single-person scenarios of *Range of Motion* (RM) and the multi-person ones in *Haggling* (HA). The RM scenarios

---

[2] https://www.idiap.ch/dataset/dih

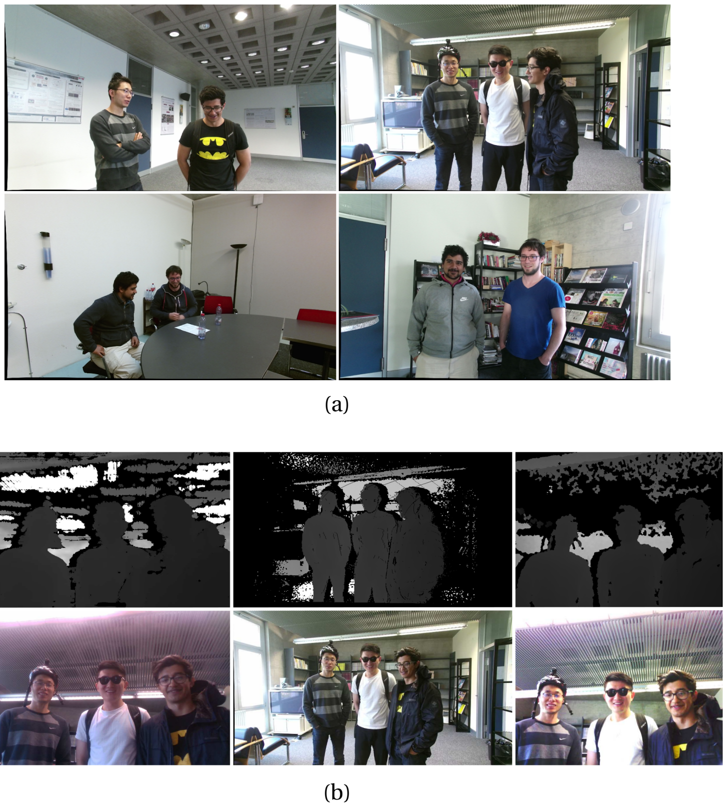
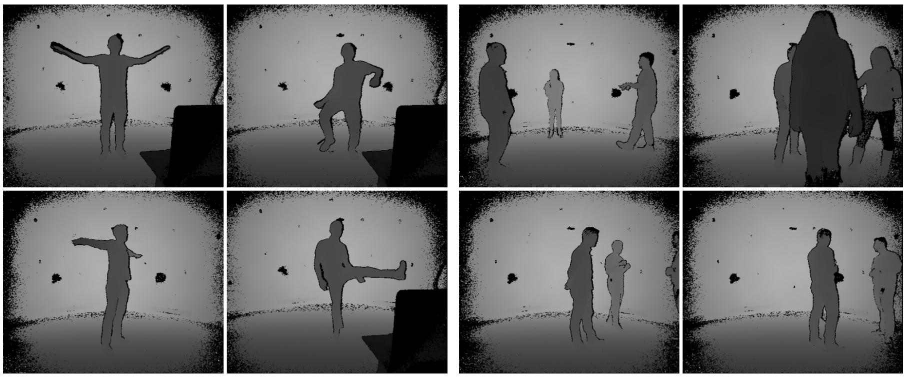


Figure 3.4 – Examples of recorded real data in HRI scenarios. (a) RGB images from the R-DIH dataset. (b) Examples of paired RGB-D images from 3 sensors. Left to right Intel D435, Kinect 2 and Asus Xtion (Pepper).

correspond to recordings of a single person performing different actions and diverse body poses. The HA scenarios consist of multi-person recording of up to 4 people playing a game where two sellers promote their own competitive products and a buyer selects one between them. Figure 3.5 shows some depth image examples.

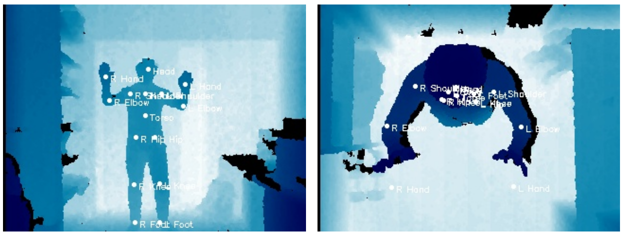
3.2.3 ITOP Pose Estimation Dataset

The ITOP dataset (Haque et al., 2016) consists on a series of single-person scenarios recorded with Asus Xtion depth cameras. It contains 50K real-world depth images from two different camera viewpoints: top-to-down view and frontal view, and 20 people performing 15 different actions. Each depth image is annotated with 3D body joint locations in the reference frame of the camera viewpoint. Some examples are shown in Figure 3.6.



(a) (b)

Figure 3.5 – Examples of the CMU-Panoptic dataset. (a) Single person range of motion scenarios; (b) Multi-person haggling scenarios.



(a) (b)

Figure 3.6 – Examples of the ITOP dataset. (a) Front view; (b) Top view.

3.3 Summary

This chapter presented the datasets of depth images that we consider in this thesis for training and evaluating our algorithms.

We introduced our large scale synthetic depth image dataset. It was generated using a randomized rendering pipeline using motion capture sequences re-targeted to a collection of 3D human characters. We add variability to the dataset by modifying the physical features of the 3D characters, randomly placing virtual cameras, and considering mocap sequences with different actions. Additionally, we take into account the multi-person scenarios by rendering images with two 3D characters. The corpus contains over 260K images of single and multi-person settings with 2D and 3D landmark annotations, as well as colored body shape masks.

We also introduced real datasets consisting of depth images recorded with depth cameras in real world scenarios. First, we described our dataset of recordings of multi-person HRI scenarios with Pepper robot. The dataset comprises 16 indoor sequences with a total of nine different participants engaging in interactions with the robot and between them. The corpus contains over 15K depth images recorded with a Kinect v2. Finally, we introduced two state-of-the-art benchmark datasets of multi-person social interactions and single-person actions.

4 Efficient 2D Multi-Person Pose Estimation

This chapter, based on (Martínez-González et al., 2020a, 2018b), discusses variouss efficient CNN models and learning approaches for fast and reliable 2D multi-person pose estimation from depth images.

As discussed in chapter 2, CNN-based approaches are the state of the art for human pose estimation. However, these methods traditionally use a deep and overparametrized architecture pretrained on a large scale image recognition dataset. This chapter addresses the limitations introduced by these design choices, aiming to reduce the computational burden with lighter CNN architectures and address the data needs for training. Firstly, we focus in depth images which are relatively simpler than color images while still containing rich and sufficient information for efficient human pose estimation. Tough some fine image details are lost, e.g. eyes and mouth, extra information important for HRI is also introduced such as scale and distance to the camera. Additionally, depth images foster 3D scene modelling and 3D pose estimation with very few assumptions compared to methods that estimate normalized 3D coordinates from color images. Secondly, inspired by efficient network structures such as those encountered in ResNets (He et al., 2016), MobileNets (Howard et al., 2017) and SqueezeNets (Iandola et al., 2016), we introduce novel lightweight network architectures that match our real-time and performance requirements.

Our proposed efficient CNN architectures are discussed in Section 4.1. In Section 4.1.1 we formalize the convolutional pose machines CNN architecture class which adopts the multi-stage prediction scheme to sequentially refine predictions as in a cascade of detectors approach. Our CNN designs are instances of this CNN class and they exploit the ResNet, MobileNet and SqueezeNet modules to reduce the computation cost of common convolutions and the number of parameters of the models (discussed in Section 4.1.2). Following the state-of-the-art method (Cao et al., 2017) our networks are trained to predict confidence maps for landmark localization and vector field for the localization and orientation of body limbs (Section 4.1.3).

Efficient and lightweight CNN models comprise much less number of parameters. As a consequence, training them from scratch using only the ground truth annotations might be harder

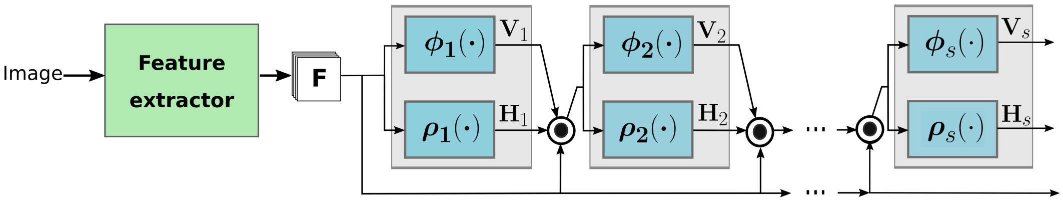


Figure 4.1 – Pose machine architecture class. It comprises a feature extractor module and a prediction cascade. Each stage in the prediction cascade is composed by two branches that predict confidence maps of body landmarks and body limbs in the image. They take as input the extracted features **F** and the confidence maps from the previous stage to refine predictions.

due to their lower learning capacities. In this thesis we further boost the generalization abilities of our lightweight models by employing knowledge distillation. Knowledge distillation (Hinton et al., 2015) aims to transfer the generalization capacities of larger and more accurate models to smaller and more efficient models. Section 4.2 discusses the use of knowledge distillation for our pose estimation settings. We employ different distillation techniques and illustrate how to couple these with our architectures designs to improve performance while maintaining efficiency (Sections 4.2.1 and 4.2.2).

Section 4.3 discusses the performance of our models and training techniques comparing them with state of the art methods. All our models are firstly pre-trained using the synthetic dataset introduced in Section 3.1. Then, we use a small set of real images for finetune our model. We evaluate the use of the synthetic images and data transformations to train our models from scratch. Additionally, we validate the use of synthetic data with more than one person to properly learn multi-person scenarios with good performance in real data.

4.1 Efficient Human Pose Estimation

This section describes the efficient CNN models we investigated for the task of body landmark localization and pose estimation. We start by introducing the pose machine architecture which forms the base of all our models. Then, we present the different instantiations that we investigated to improve efficiency while maintaining accuracy.

4.1.1 Pose Machines CNN Architecture Class

The pose machine architecture comprises two main components: a feature extractor module and a cascade of predictors that output confidence maps for each of the body landmarks and body limbs. Figure 4.1 sketches the architecture class concept and its main components.

More precisely, the CNN takes an image as input and the feature extractor module computes an abstract representation of it composed of N_w channels, denoted as **F**. These features **F** are passed to the cascade of predictors, composed of a series of prediction stages sequentially stacked. Each prediction stage aims at localizing body landmarks (neck, elbows, ankles) and limbs, which are segments between two landmarks according to the skeleton shown in Figure 3.2 (forearms or thighs).

Each stage s consists of two branches made of fully convolutional layers predicting confidence maps of body landmarks, denoted $\rho_s(\cdot)$, and of body limbs, denoted $\phi_s(\cdot)$. For $s \geq 2$ these branches take as input both the features $\mathbf{F}$ and the landmark and limbs predictions maps from stage $s-1$. In effect, this allows the refinement of the predictions of each element (landmark and limbs) by incorporating context from the other body parts and hence accounting for valid body pose configurations. This effectively reduces the number of pairs of detected body landmarks and potentially connected, easing the single and multi-person pose inference stage.

4.1.2 Efficient Pose Machines

For an efficient forward pass, instances of the above architecture class will incorporate lightweight designs in the feature extractor, $\rho_s(\cdot)$ and $\phi_s(\cdot)$. In Figure 4.2 we illustrate the design of our efficient pose machine instances. Modules enclosed by doted squares are the components which are replicated to achieve the cascade of predictors. We describe these architectures in the following.

Residual Pose Machines

In the pose machine architecture instance presented in (Cao et al., 2017), the first computational bottleneck is the large VGG-19 architecture used as feature extractor module. Therefore, we propose to investigate how to exploit a lighter module built upon residual modules (or blocks) (He et al., 2016). We originally introduced this modification in (Martínez-González et al., 2018b). Our motivation is that residual blocks are known to outperform VGG networks, and to be faster by having a lower computational cost (Canziani et al., 2016). Figure 4.2(a) depicts the architecture we dubbed as *residual pose machines*.

Feature Extraction Network. It consists of an initial convolutional layer followed by three residual modules with small kernel sizes (3 × 3). The network has three average pooling layers. Each residual module consists of two convolutional layers and a shortcut connection. Batch normalization and ReLU are included after all convolutional layers and shortcut connections as exemplified in Figure 4.3(a).

Pose Regression Cascade. We maintain a large effective receptive field in the design of the branches $\phi_s(\cdot)$ and $\rho_s(\cdot)$. In the first prediction stage the network has three convolutional layers with filters of 3 × 3 and two layers with filters of 1 × 1, whereas in the remaining stages there are five and two convolutional layers with filters of 7 × 7 and 1 × 1 respectively.

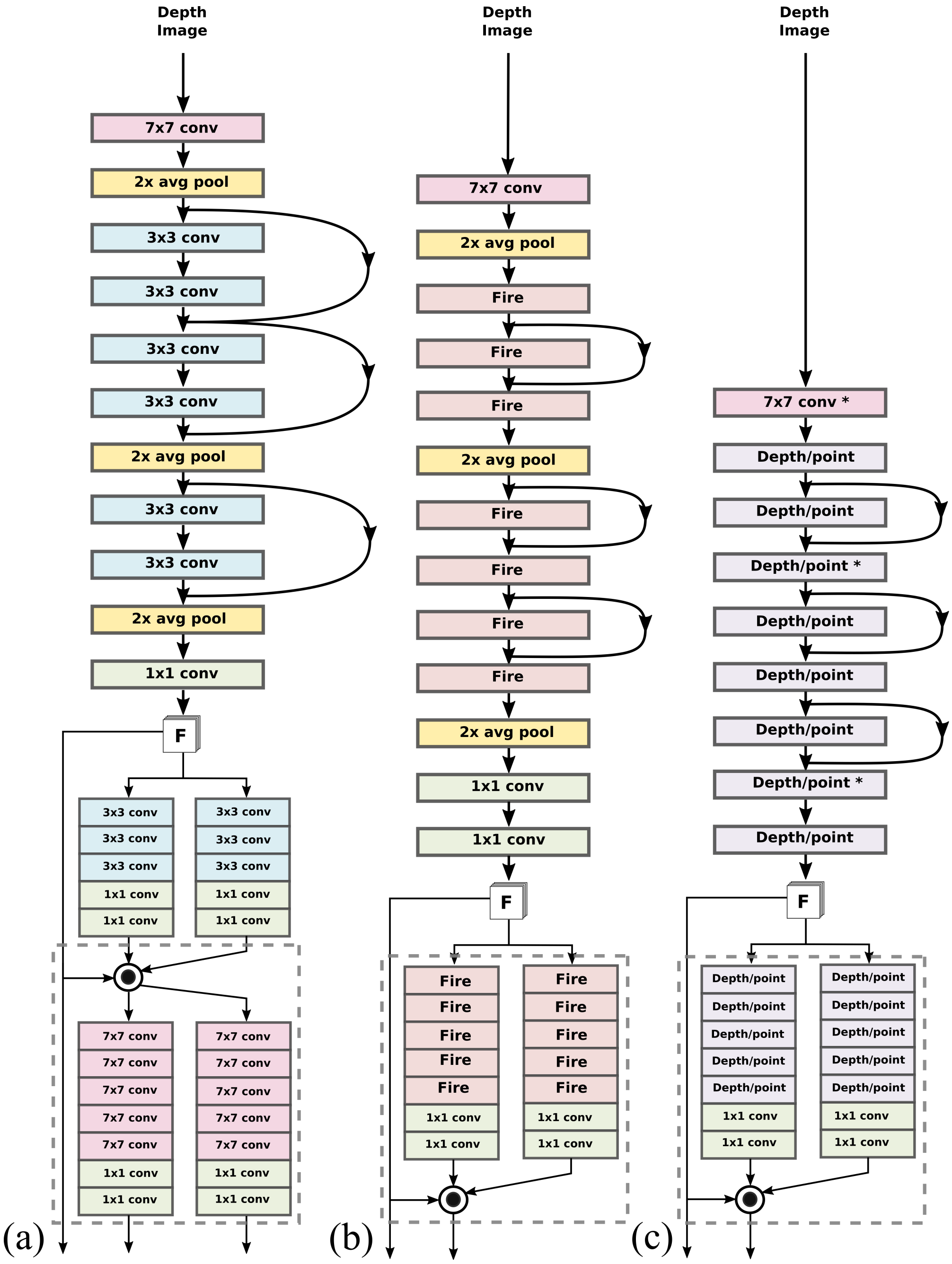


Figure 4.2 – CNN architecture instances of the pose machines class. (a) Residual pose machines focuses on speeding up the feature extractor module using ResNet modules; (b) SqueezeNet pose machines builds on the *Fire* module concept to design a lighter architectures; (c) MobileNet pose machines relies on depthwise and pointwise convolution layers to speed up computation. Convolution layers marked with * have a stride of 2 and serve as pooling mechanism.

Squeezenet Pose Machines

In (Iandola et al., 2016) the SqueezeNet architecture is build upon a series of modules called *Fire* modules. Each module is composed by a *squeeze* layer and an *expand* layer. The squeeze layer contains filters of 1×1 and outputs N_{s_1} channels, while the expand layer is a mix of filters of 1×1 and 3×3 that outputs N_{e_1} and N_{e_3} channels respectively. This configuration aims to reduce the model size by using 1×1 filters, and to speed up computation by limiting the number of input feature channels of the 3×3 layers. Figure 4.3(b) illustrates the composition of a Fire module.

Our architecture design using Fire modules is shown in Figure 4.2(b). The output of the Fire module is the concatenation of the channels from the 1×1 and 3×3 expand layers. We follow the original design and set N_{s_1} to be less than the sum $N_{e_1} + N_{e_3}$ and maintain these quantities low. Batch normalization and ReLU are added after each convolution.

Feature extraction network. The feature extractor module comprises 7 Fire modules, three average pooling layers, three residual connections and two 1×1 convolution layers. ReLU activations are included after the shortcut connection following the design in Figure 4.3(a).

Pose regression cascade. The pose regression cascade employs five Fire modules and two 1×1 convolution layers in the design of both branches $\phi_s(\cdot)$ and $\rho_s(\cdot)$.

Mobilenet Pose Machines

MobileNets (Howard et al., 2017) are built on separate filters factorizing a standard convolution layer into a *depthwise* and a *pointwise* convolution layers. A depthwise operation applies a spatial filter to each input channel (one different filter per channel). Pointwise filters are the classical 1×1 convolution filter that performs a linear combination of all channels of the depthwise output. This factorization has a high impact in the size and the computation that the model requires. Figure 4.3(c) illustrates the design we follow to implement depthwise and pointwise convolution filters. 3×3 filters are used for the depth wise convolutions. We add batch normalization and ReLU units after each convolution.

Feature extraction network. It is composed of 8 depthwise - pointwise layers, denoted as Depth/point in Figure 4.2(c). Pooling mechanisms are included in the form of convolutions with stride 2 (denoted with *). We also include three residual connections following the design in Figure 4.3(a).

Pose regression cascade. In each stage, we compose both branches by 5 depthwise-pointwise modules followed by two 1×1 convolution layers.

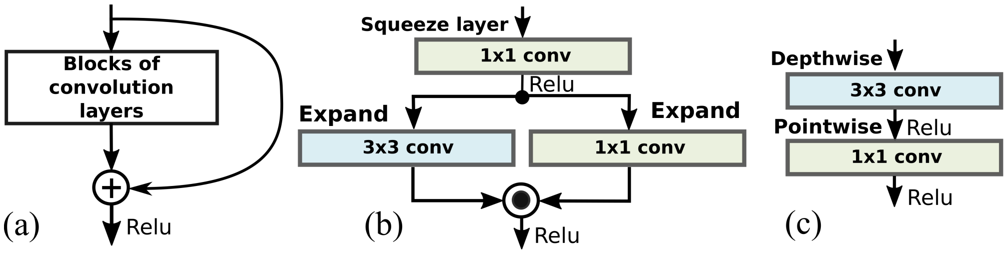


Figure 4.3 – Different unit modules used in our architecture designs. (a) In the residual module we use the sum operation to combine inputs from the shortcut connection and the set of convolution layers; (b) fire modules in SqueezeNets combines 1 × 1 and 3 × 3 convolution layers output with a concatenation operation; (c) in MobileNets a standard convolution is decomposed into depthwise and pointwise convolution layers.

4.1.3 Confidence Map Ground Truth and Training

We regress confidence maps for the location of the different body parts and predict vector fields (part affinity fields) for the location and orientation of the body limbs. The ideal representation of the body part confidence maps $\mathbf{H}^*$ encodes their locations in the depth image as Gaussian peaks. Let $\mathbf{x}_j$ be the ground truth position in the image of body part j. The value $\mathbf{H}^*_j(\mathbf{p})$ for pixel $\mathbf{p}$ in the target confidence map $\mathbf{H}^*_j$ of the jth part is computed as

$$\mathbf{H}^*_j(\mathbf{p}) = exp\left(-\frac{||\mathbf{p}_j - \mathbf{x}_j||_2^2}{\sigma^2}\right), \tag{4.1}$$

where σ is chosen empirically.

Regarding the limbs, the ideal representation $\mathbf{V}^*$ encodes with a vector field the confidence that two body parts are associated, as well as the information about the orientation of the limbs. Consider a limb of type c that connects two body parts j_1 and j_2 (e.g. elbow and wrist) and their positions on the depth image are $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$. The ideal affinity field at point $\mathbf{p}$ is defined as

$$\mathbf{V}^*_c(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ is on limb } c \\ 0 & \text{otherwise} \end{cases}, \tag{4.2}$$

where $\mathbf{v}$ is the unit vector that goes from $\mathbf{x}_{j_1}$ to $\mathbf{x}_{j_2}$. The set of pixels that lie on the limb are those within a distance to the line segment that joins the two body parts.

Training. Supervision is applied at the end of each prediction stage to prevent the network from vanishing gradients. This supervision is implemented by two L_2 loss functions, one for each of the two branches, between the predictions $\mathbf{H}_s$ and $\mathbf{V}_s$ and the ideal representations $\mathbf{H}^*$

and $\mathbf{V}^*$ for stage s

$$L_s^{\mathbf{V}} = \sum_{\mathbf{p}\in\mathbf{I}} ||\mathbf{H}_s(\mathbf{p}) - \mathbf{H}^*(\mathbf{p})||_2^2, \tag{4.3}$$

$$L_s^{\mathbf{H}} = \sum_{\mathbf{p}\in\mathbf{I}} ||\mathbf{V}_s(\mathbf{p}) - \mathbf{V}^*(\mathbf{p})||_2^2. \tag{4.4}$$

The final multi-task loss is computed as:

$$L_{PM} = \sum_{s=1}^{S} \left(L_s^{\mathbf{H}} + L_s^{\mathbf{V}}\right), \tag{4.5}$$

where S is the total number of prediction stages.

4.1.4 Body Part Association

We use the algorithm presented in (Cao et al., 2017) that uses the part affinity fields as confidence to associate the different body landmarks and perform the inference of the 2d pose. It works in a greedy fashion exploiting the skeleton tree structure, analyzing pairs of body landmarks that are potentially linked by a limb type and builds the pose estimate increasingly.

In a nutshell, the method takes as input a given limb type, e.g. forearm. All detected body landmarks that form such limb type (wrists and elbows) are connected one to one, forming potential limbs. The connections are weighted by averaging the predicted vector field along the line that joins the pair of body landmarks. Connections with high confidence are kept while the other discarded. Final pose estimates are built by associating the limb candidates (forearm and arm) that share body landmark candidates (shoulder, elbow and wrist).

4.2 Distilling The Pose From a Network

Training a CNN model usually relies on data with ground truth. However, the data samples are often of different complexity levels. Intuitively, factors like the camera view point, the pose configuration, the occlusion level (whether due to external elements or to its own body), clothing or object artefacts will render the landmark prediction more or less difficult, an element which is not reflected in the ground truth. While large over-parametrized networks, when supplied with enough data, can usually accommodate this diversity during training, smaller more efficient networks, by attempting to satisfy the ground truth of all data equivalently, can be more easily trapped in local minima with limited generalization. Several methods have been proposed to improve training. One of them is curriculum learning Bengio et al. (2009), i.e. starting from easier to more complex data samples. Another direction is knowledge distillation that we now introduce.

The distillation approach can be posed as follows: given a deep and large teacher network

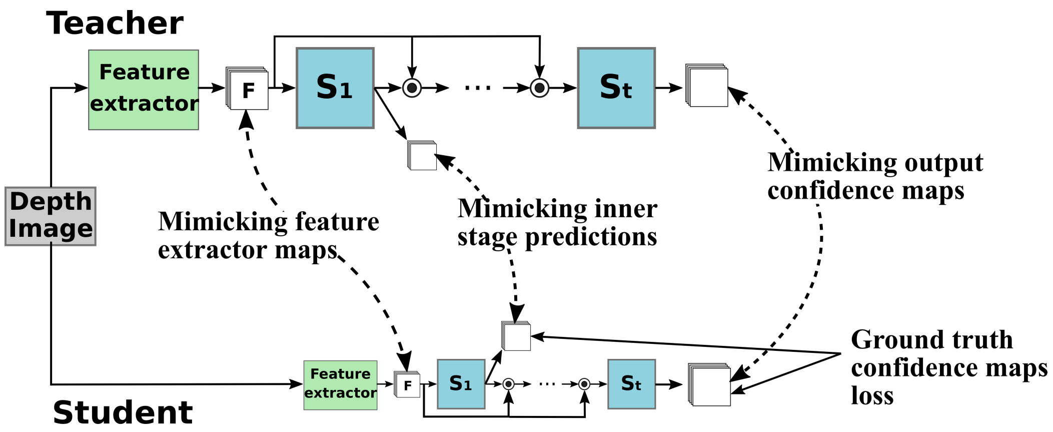

Teacher
Feature extractor
F
S1
St
Depth Image
Mimicking feature extractor maps
Mimicking inner stage predictions
Mimicking output confidence maps
Ground truth confidence maps loss
Feature extractor
F
S1
St
Student

Figure 4.4 – Knowledge distillation scheme for body landmark detection in a cascade of detectors fashion. The student is trained to mimic (in addition to ground truth) the confidence map predictions of body landmarks and body limbs from a pre-trained and robust teacher at the different stages in the cascade of regressors. We additionally adopt the learning by hints approach by encouraging the features of student to match those of the teacher.

T_{net}, we would like to improve the generalization ability of an efficient student S_{net} using the "knowledge" acquired by T_{net}. Such knowledge transfer can be of several forms. First, it usually involves mimicking the output activations of the teacher as these activations implicitly contains information about the difficulty of the samples (e.g. ambiguity between classes, uncertainty due to difficult imaging conditions). However, it can also include mimicking some of the hidden layers activation maps. In this case, the activations are referred as *hints* and the goal is to drive the student learning towards learning intermediate representations thought as important from a design process.

We learn accurate and efficient models by using a teacher with high performance using the distillation strategy illustrated in Figure 4.4. We have investigated several configurations. First, performing distillation at every stage on the cascade of the teacher, i.e. matching the predictions of the teacher at every prediction stage in the cascade to distil the knowledge at every prediction stage and to promote an early on semantic knowledge distillation. Note that this contrasts with the conventional distillation approach which considers only the final prediction as knowledge to transfer. Second, adopting distillation by hints to encourage the student to learn a data representation similar to that of the teacher. These two approaches can be combined in an overall pose distillation objective written as follows:

$$L_{distil} = L_{stages} + \gamma L_{hints}. \tag{4.6}$$

The L_{stages} and L_{hints} losses are described below.

4.2.1 Mimicking Teacher Stage Predictions

We couple knowledge distillation with our architecture designs and perform distillation at the last prediction stages of the cascade. The motivation is that the teacher's predictions in these last stages contain valuable information about how the teacher refines predictions, which may help the student on how to increasingly improve its own predictions. In practice, we use a weighted sum of losses considering the teacher's predictions and the ground truth L_{PM} which also introduces information that the student should mimic:

$$L_{stages} = (1-\alpha)L_{teacher} + \alpha L_{PM}, \tag{4.7}$$

where L_{PM} is defined in Eq (4.5) and α is a weighting parameter set to 0.5 in our experiments. We choose to model $L_{teacher}$ as to match the prediction of the τ ($\tau \geq 1$) last stages of the cascade. $L_{teacher}$ is defined as

$$L_{teacher} = \sum_{s=0}^{\tau-1} \left(L_{S-s}^{\mathbf{H}} + L_{S-s}^{\mathbf{V}} \right), \tag{4.8}$$

where S is the number of prediction stages in the teacher's cascade, and

$$L_s^{\mathbf{H}} = \sum_{\mathbf{p} \in \mathbf{I}} ||\mathbf{H}_s(\mathbf{p}) - \mathbf{H}_s^{tea}(\mathbf{p})||_2^2, \tag{4.9}$$

$$L_s^{\mathbf{V}} = \sum_{\mathbf{p} \in \mathbf{I}} ||\mathbf{V}_s(\mathbf{p}) - \mathbf{V}_s^{tea}(\mathbf{p})||_2^2, \tag{4.10}$$

where $\mathbf{p}$ is a pixel in image $\mathbf{I}$, and ($\mathbf{H}_s^{tea}$,$\mathbf{V}_s^{tea}$) are the body part and part affinity fields confidence maps at stage s of the teacher. Notice that the number of stages in the teacher and student prediction cascades need to be at least τ.

4.2.2 Learning With Hints

We consider this approach by enforcing similarities between the feature extractor outputs in the teacher and student architectures, as it is a natural choice in our architecture designs. Note that our formulation in Eq (4.8) is a form of distillation with hints for $\tau > 1$.

Thus, denoting by $\mathbf{F}_{\mathbf{I}}^{tea}$ the internal representation of image $\mathbf{I}$ of the teacher, we define the hints loss as

$$L_{hints} = ||\mathbf{F}_{\mathbf{I}} - \mathbf{F}_{\mathbf{I}}^{tea}||_2^2, \tag{4.11}$$

Note that when performing knowledge distillation with hints it is necessary to tie the dimensions of $\mathbf{F}_{\mathbf{I}}$ and $\mathbf{F}_{\mathbf{I}}^{tea}$.

4.3 Experiments

In this section we describe the experiments we performed to evaluate our approaches. The experimental protocol focuses on both accuracy and computational aspects. We first evaluate the proposed efficient CNN architectures and the impact of the synthetic dataset for training. Finally, we analyze our approach for knowledge distillation and its benefits in our context.

4.3.1 Data

We considered the synthetic and real parts of the publicly available DIH dataset introduced in Sections 3.1.3 and 3.2.1, as well as a subset of the CMU-Panoptic dataset, both comprising Kinect 2v images. Although the datasets contain color images as well, we focus our analysis on depth images and leave the exploitation of color data for another study. With both synthetic and real images, we performed data augmentation during training: rotation by a random angle within $[-20, 20]$ degrees with a 0.8 probability, and image cropping to the 368×368 training size with a probability of 0.9. Unless stated otherwise, we use R-DIH as our default (real) dataset in the result section.

Synthetic Data

S-DIH train, validation and testing folds contain 230934, 22333 and 11165 synthetic depth images respectively. To avoid our pose detector to overfit clean synthetic image details, we propose to add image perturbation, in particular, adding real background content which will provide the network with real sensor noise.

Adding Real Background Content. We fuse real background images with our synthetic data. We produce training images on the fly by randomly augmenting synthetic depth images fusing them with real depth background images as described in Section 3.1.4.

Pixel Noise. During training we randomly selected 20% of the body silhouette's pixels and set their value to zero.

Real Data

We consider the following datasets for evaluation and fine-tuning.

R-DIH. It consists of 16 sequences divided into train, validation and testing folds comprising 7, 5 and 4 sequences respectively. We manually annotated a small subset of images for each fold: 1750, 750 and 1000 images within the training, validation, and testing folds, respectively.

Panoptic. We used a subset of the large CMU-Panoptic dataset (Joo et al., 2017) from the *Range of Motion (RM)* and *Haggling (H)* scenarios (specifically, RM:171204_pose3 and H:170407_haggling_a3 for training, RM:171204_pose5 and H:170407_haggling_b3 for testing). To be consistent with our experimental setting, where few labeled data are available, we randomly considered 2K images for training and 1K for testing.

4.3.2 Evaluation metrics

Pose Estimation Performance. We use standard precision and recall measures derived from the Percentage of Correct Keypoints (PCK) evaluation protocol as performance metrics (Yang and Ramanan, 2013). More precisely, after the forward pass of the network, we extract all the landmark predictions p whose confidence are above a threshold η, and run the part association algorithm to generate pose estimates from these predictions[1]. Then, for each landmark type, and for each ground truth points q, we associate the closest prediction p (if any) whose distance to q is within a distance threshold $d = \kappa \times h$, where h stands for the height of the bounding box of the person (in the ground truth) to which q belongs to. Such associated p are then counted as true positives, whereas the rest of the landmark predictions are counted as false positives. Ground truth points q with no associated prediction are counted as false negatives. Recall and precision values are computed for each landmark type counting true positives, false positives and false negatives over the dataset. Finally, the average recall (AR) and average precision (AP) values used to report performance are computed by averaging the above recall and precision over landmark type and over several distance thresholds d by varying $\kappa \in [0.05, 0.15]$.

Computational performance. Model complexity is measured via the number of parameters it comprises, and the number of frames per second (FPS) it can process when considering only the forward pass of the network. This was measured using the median time to process 2K images at resolution 444×368 pixels with an Nvidia card GeForce GTX 1050.

4.3.3 Implementation Details

Image Preprocessing. The depth images are normalized by scaling linearly the depth values in the $[0, 8]$ meter range into the $[-0.5, 0.5]$ range.

Network training. Pytorch is used in all our experiments. We train different network architectures with stochastic gradient descent with the momentum set to 0.9, the decay constant to 5×10^{-4}, and the batch size to 10. We uniformly sample values in the range $[4 \times 10^{-10}, 4 \times 10^{-5}]$ as starting learning rate and decrease it by a factor of 10 when the validation loss has settled. All networks are trained from scratch and progressively, i.e. to train network architectures with s stages, we initialize the network with the parameters of the trained network with $s - 1$ stages. Unless otherwise stated, each model was trained with synthetic data for 13 epochs and

---

[1] Note that in this algorithm, landmark keypoints not associated with any estimates are automatically discarded.

DIH										
					All body			Upper body		
Architecture	# Stages	# Features	# Parameters	FPS	AP	AR	F-Score	AP	AR	F-Score
HG (Newell et al., 2016)	2	-	12.93 M	8.7	84.62	86.26	0.85	86.45	93.70	0.90
CPM-1S (Cao et al., 2017)	1	128	8.38 M	18.6	92.10	86.44	0.89	96.12	94.99	0.95
CPM-2S (Cao et al., 2017)	2	128	17.07 M	11.2	94.03	88.96	**0.91**	96.36	94.98	**0.96**
RPM-1S	1	64	0.51 M	56.7	90.86	72.07	0.80	94.28	87.77	0.91
RPM-2S	2	64	2.84 M	35.2	94.84	86.41	0.90	96.39	93.91	0.95
RPM-3S	3	64	5.17 M	20.8	93.96	87.72	**0.91**	97.26	94.72	**0.97**
RPM-2S	2	128	10.5 M	12.5	93.52	86.07	0.90	95.95	94.02	0.95
MPM-1S	1	64	99.8 K	134.4	88.52	71.36	0.79	92.88	84.74	0.89
MPM-2S	2	64	168.2 K	112.3	92.56	78.63	0.85	94.97	89.92	0.92
MPM-3S	3	64	236.5 K	95.8	92.40	82.79	0.87	95.23	92.36	0.94
MPM-4S	4	64	304.9 K	84.3	91.27	84.06	**0.88**	95.27	91.87	**0.94**
SPM-1S	1	64	308.8 K	70.6	89.64	61.58	0.73	93.18	71.38	0.81
SPM-2S	2	64	455.9 K	60.6	91.78	81.68	0.86	95.82	91.47	0.94
SPM-3S	3	64	660.0 K	53.3	92.63	81.98	**0.87**	96.43	90.84	**0.94**
SPM-4S	4	64	921.0 K	47.2	93.13	81.36	0.87	96.30	90.58	0.93

Panoptic										
					All body			Upper body		
Architecture	# Stages	# Features	# Parameters	FPS	AP	AR	F-Score	AP	AR	F-Score
HG (Newell et al., 2016)	2	-	12.93 M	8.7	85.14	91.06	0.88	86.0	91.0	0.88
CPM-2S (Cao et al., 2017)	2	128	17.07 M	11.2	96.73	91.66	**0.94**	96.75	92.07	**0.94**
RPM-3S	3	64	5.17 M	20.8	97.42	91.48	**0.94**	97.43	91.48	**0.94**
MPM-4S	4	64	304.9 K	84.3	96.43	89.17	0.92	97.0	89.06	0.93
SPM-3S	3	64	660.0 K	53.3	95.44	89.24	0.92	96.30	90.35	0.93

Table 4.1 – Performance (%) on the test set of real depth images and architecture components for the different tested pose machines instances. Upper body comprises only head, neck, shoulders, elbows and wrists.

finetuned with real images for 100 epochs.

Tested Models and Notation. Residual, mobile and squeeze pose machines are referred to as RPM, MPM and SPM respectively. We set $N_w = 64$ (number of features) as the default value and we specify when it changes. We add a postfix to specify the number of stages that a model comprises. For example, RPM-2S is the residual pose machine configuration with 2 prediction stages in the cascade of predictors.

4.3.4 Performance-Efficiency Trade-Off

Table 4.1 compares the performance of our proposed architecture configurations. We report both the average recall and average precision for all landmark types in the skeleton model and for the upper body, i.e. *head, neck, shoulders, elbows and wrists* since upper body detection might be sufficient for any given applications (e.g. HRI). We compare the models' performance with the F-Score metric (harmonic mean of recall and precision). The table also compares the FPS and the number of trainable parameters of the different architectures.

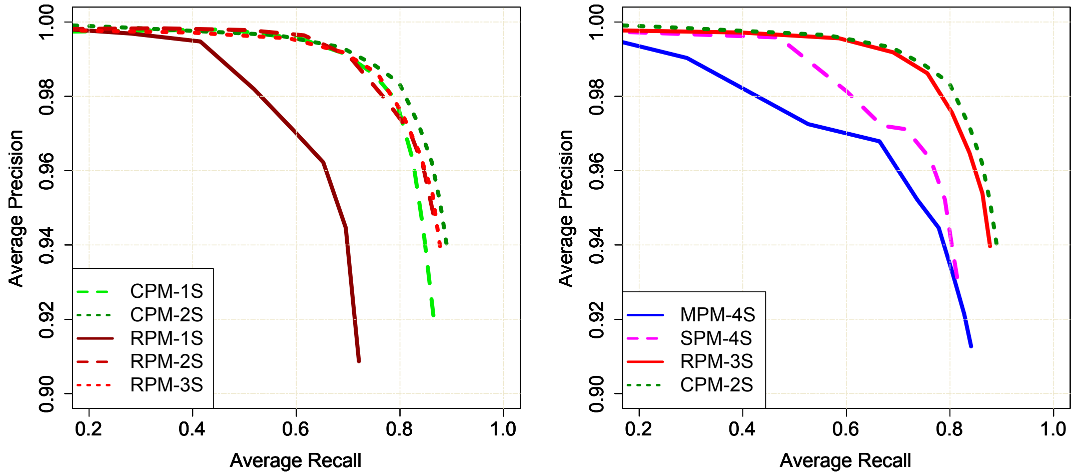


Figure 4.5 – Precision-recall curves obtained by varying the landmark detection threshold η. Left: comparison between the baseline CPM and the RPM model with different number of prediction stages. Right: performance comparison of our pose machines instances in their deepest version.

Comparison with The Convolutional Pose Machine (CPM) Baseline

We consider the CPM architecture presented in (Cao et al., 2017) as the main baseline. As in the original work, the architecture parameters in the feature extractor module were initialized using the first 10 layers of the VGG-19 network. Parameters in the cascade of detectors were trained from scratch. To accommodate the need for the 3 channel image input expected by VGG-19, the single depth channel is repeated three times. We report the results for architectures comprising up to two stages in the prediction cascade since no substantial performance increase was obtained after the second stage.

We train RPM network architectures with 1, 2 and 3 stages. Figure 4.5(a) shows the precision-

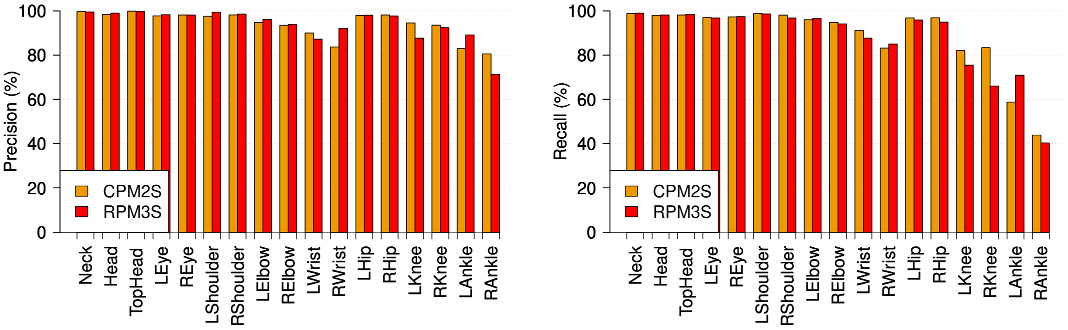


Figure 4.6 – Precision (left) and recall (right) per body landmark for the RPM-3S and the baseline CPM-2S.

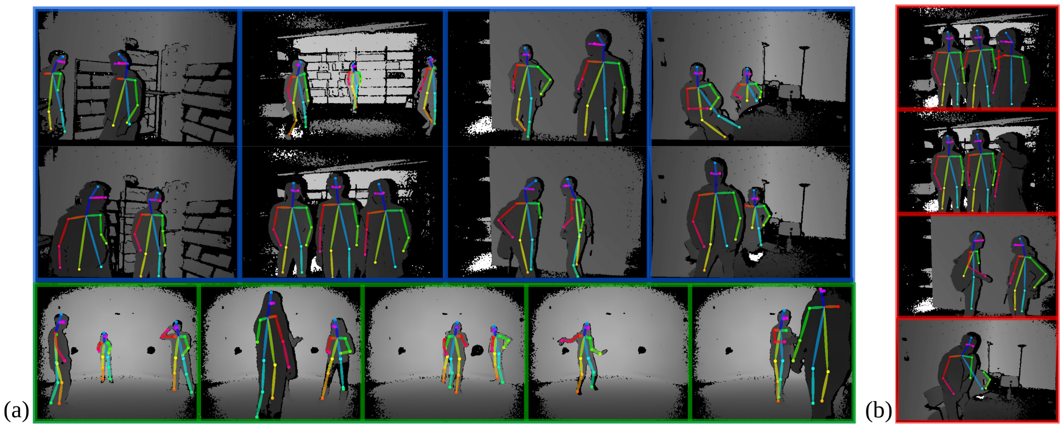


Figure 4.7 – Left: Pose estimation examples from the RPM-3S landmark detector on the DIH dataset (blue squares) and CMU-Panoptic dataset (green squares). Right: Examples of failures cases. They often occur for specific pose or clothings or under occlusion of people or objects (e.g. bags).

recall curves over the testing set of real depth images, obtained by varying the landmark detection threshold η. We summarize these curves in Table 4.1 taking the performance with the highest F-Score. As can be seen, our RPM models perform as well as the baseline but is lighter and faster. This is specially the case for RPM-2S that shows similar performance as CPM-2S but comprises 6 times less parameters and is 3.14 times faster. Interestingly, we can notice from Figure 4.5(a) that the smaller complexity of the feature extractor of RPM-1S leads to degraded performance compared to the baseline CPM-1S. This gap is filled once context is introduced with the second stage (see RPM-2S and CPM-2S curves). Adding an extra stage (RPM-3S) slightly improves the model performance while still being faster than the baseline. In particular for the upper body parts where it now slightly outperforms the baseline. Figure 4.6 compares the per body landmark precision and accuracy for RPM-3S and CPM-2S models, where we can notice very high performance achieved for the upper body landmarks.

Number of feature channels. We set $N_w = 128$ and train the RPM models. We report the results for RPM-2S with this configuration in Table 4.1. Note how the results of the RPM-2S with $N_w = 64$ and RPM-2S with $N_w = 128$ are very similar showing that more feature channels does not bring more benefits. Given its higher computational complexity setting $N_w = 64$ is a good accuracy-speed trade-off.

Qualitative analysis. Figure 4.7 shows examples of the pose estimation algorithm using the body landmarks and limbs output of our RPM-2S model. Note that our model is capable of producing strong confidence maps that produce accurate estimates even for the eyes and in the presence of self and person occlusions, profile views, and different body pose configurations and silhouette shapes. The main challenges for our model include strong changes in the person silhouette (backpacks, big jackets) and person proximity and occlusions as illustrated by failure cases in Figure 4.7(b).

Efficient Pose Machines

The performance of the MPM and SPM models are presented in Table 4.1. We can first notice that they are more efficient in terms of FPS and number of trainable parameters than the CPM and RPM models. For example, MPM-4S contains 55.9 times less parameters than CPM-2S model and is 7.5 times faster with only a decrease of 0.03 in F-Score. SPM-4S, on the other hand is 4.2 times faster than CPM-2S with a decrease of 0.04 in F-Score.

We note that increasing the number of prediction stages improves the models' F-Score. As with the RPM models, we observe that the biggest improvement appears when introducing a second stage. The additional stages help refining prediction, for instance by greatly improving the landmark detection recall while the precision saturates or slightly degrades (compare the MPM-2S and MPM-4S results), but in general the results often start saturating after the 3rd stage.

Figure 4.5(b) illustrates the precision-recall curves for the deepest version of our efficient models. Among the architectures we investigate, the fastest are the MPM designs, followed by SPM models. Nevertheless, the best speed-accuracy trade-off are given by the RPM-2S model when the focus is on accuracy, and MPM-4S when it is on speed. A visualization of such efficiency-accuracy trade-off is given in left plot of Figure 4.8.

Comparison with State-Of-the-Art Methods

In addition to the CPM baseline, we compared our methods with the stacked Hourglass framework (Newell et al., 2016). We used the same network architecture and training protocol (initialization, learning rate, optimizer) proposed by the authors. For a fair comparison we followed our protocol regarding the data, trained the model with synthetic images for 13 epochs and finetuned it with real data for 100 epochs.

Results in Table 4.1 show that our efficient architectures outperform this baseline in efficiency and accuracy, on both the DIH and Panoptic datasets. For example, MPM-4S is 9.6 times faster, 42.2 times smaller, with an F-Score of 0.88 (on DIH) compared to 0.85 for the Hourglass network.

Experiments with CMU-Panoptic Dataset

The results on this dataset are shown in Table I (bottom part). We report only the results obtained by the best performing models on the DIH-Real dataset. Our proposed RPM-3S outperforms the Hourglass baseline. It also performs similarly than the CPM baseline. The proposed efficient architectures MPM-4S and SPM-3S outperform the Hourglass baseline by a margin of 0.04 in the F-Score and follow closely the RPM-3S and baseline CPM-2S with a difference of 0.02 in the F-Score.

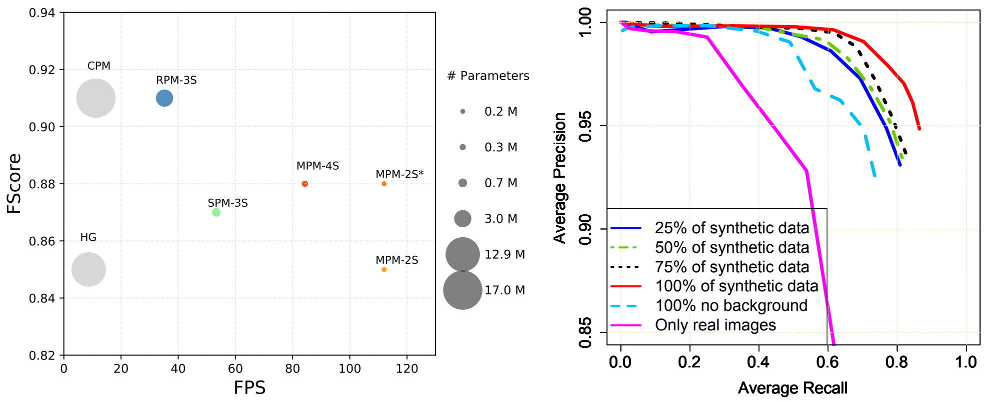


Figure 4.8 – Left: efficiency-accuracy tradeoff visualization for the best models on DIH-Real. Models marked with * have been trained with knowledge distillation. Right: evaluation of the use of synthetic data for learning robust pose estimation models.

	Synthetic only			Synthetic + Finetuning		
Synthetic data %	AP	AR	F-Score	AP	AR	F-Score
25 %	94.19	26.28	0.41	93.09	80.83	0.86
50 %	95.10	29.51	0.45	93.45	81.38	0.87
75 %	91.68	30.14	0.45	93.68	82.42	0.88
100 %	90.37	31.14	0.46	**93.96**	**87.72**	**0.91**
100 % no BG	80.67	3.35	0.06	92.58	73.46	0.82

Table 4.2 – Comparison of performance obtained with RPM-2S when trained with different synthetic data partitions.

4.3.5 Training with Synthetic Data Analysis

We validate the use of the synthetic data to learn accurate models. To this end, we randomly split the synthetic training data in partitions comprising 25%, 50%, 75% and 100% of it. These partitions contain images with one and two people. We train RPM-2S models with the different synthetic data partitions and then finetune the result with real data. We use the same quantity of labeled data to finetune in all cases (1750 images). Unless otherwise stated, during training we consider all image transformations for data augmentation (background fusion, pixel drop, cropping and rotation). Table 4.2 shows the obtained average recall and precision on real data, before and after finetuning. The following conclusions can be made.

Amount of Synthetic Data

Performance increases with more synthetic data, both before and after finetuning. Naturally, the visual features mismatch between the synthetic and real data provokes low performance when training only with synthetic data. Nevertheless the gap is covered once finetuning on real data is applied, particularly regarding the recall. Figure 4.8(b) shows the average

	All test data			Data with person occlusions		
Data fold	AP	AR	F-Score	AP	AR	F-Score
1P-Fold	92.51	80.22	0.86	91.10	82.92	0.87
2P-Fold	92.43	82.44	**0.87**	93.33	84.81	**0.89**

Table 4.3 – Performance obtained with the RPM-2S model when combining synthetic data with only single person images (1P-Fold) and 2 people images (2P-Fold) for training.

precision-recall curves training with the different data partitions and applying finetuning.

Adding Realism to Synthetic Images

We validate our strategy of fusing real background with synthetic depth images to prevent overfiting. To this end, we trained the RPM-2S models with 100% of the synthetic data, helding out the background fusion transformation, and then applied finetuning. We report the results in Table 4.2. Observe that without the addition of real background, our model overfits to the synthetic data details and performs poorly on real data. Interestingly, note as well that even after finetuning on real data the performance is not entirely recovered. The model even performs lower than using only 25% of the synthetic data. Intuitively, fusing background with synthetic images work as a regularizer that prevents overfitting to synthetic image details.

Multiple People Data

We study the importance of having synthetic training images with two people before finetuning with real data. To this end, we define two folds of 100K images for training. The first one (1P-Fold) contains only images with one person; the second one (2p-Fold) contains 50K images with one person and 50K images with two people. The resulting performance is reported in Table 4.3 where we also provide results on a test subset where people are very close or occlude each other (see Figure4.7(b)). The subset contains 211 images with three people where their ground truth bounding boxes overlap between 12.38% and 15.4%. We note that using images with two people helps generalization.

Training with Real Data Only

We train the RPM-2S model only with our small real depth image annotated sample. Figure 4.8(b) shows the performance curve. Our real dataset sample is not large enough to prevent our model from overfiting and performs worse than using the synthetic data without background fusion.

Student	Teacher	Distil type	AP	AR	F-Score
DIH					
CPM-2S	-	-	94.03	88.96	0.91
MPM-2S	-	-	92.56	78.63	0.85
MPM-4S	-	-	91.27	84.06	0.88
MPM-1S	CPM-2S	Stagewise	88.55	71.96	0.79
MPM-1S	CPM-2S	Stagewise + Hints	89.89	76.61	0.83
MPM-2S	CPM-2S	Standard ($\tau = 1$)	90.98	80.60	0.85
MPM-2S	CPM-2S	Stagewise	**92.10**	**83.98**	**0.88**
MPM-2S	CPM-2S	Stagewise* ($\tau = 2$)	92.42	81.60	0.87
MPM-2S	CPM-2S	Stagewise + Hints	90.91	80.19	0.85
Panoptic					
CPM-2S	-	-	96.73	91.66	0.94
MPM-2S	-	-	96.25	86.66	0.91
MPM-4S	-	-	96.43	89.17	0.92
MPM-2S	CPM-2S	Stagewise	95.08	90.33	0.92

Table 4.4 – Knowledge distillation experiments. Methods with - in the 'Teacher' and 'Distil type' fields indicate that the model has been trained without distillation and their values were taken from Table 4.1 (i.e. these are the baseline results).

4.3.6 Pose Knowledge Distillation

We perform knowledge distillation using CPM-2S as the teacher. We select MPM-1S and MPM-2S models as students given that the MPM design is the most efficient of our pose machines. Our students are trained from scratch. We performed distillation first with synthetic data during 10 epochs, then with real images for 100 epochs. The learning rate is updated in a linear fashion and we set γ to 1. Table 4.4 reports the results. Our proposed approach of performing distillation at the last stages in cascade is referred to as *stagewise*. We also experiment by matching only the final activation maps of the teacher at every stage of the student (marked with *).

Knowledge Distillation Approaches

Our stagewise approach outperforms the same network configurations trained without distillation or with the standard distillation approach (only taking into account the last stage of the cascade). Learning with hints makes the student MPM-1S outperform its stagewise distillation counterpart, but this is not the case with MPM-2S. One possible reason is that given the architecture differences, enforcing feature similarity prevents the student from improving the task loss. In contrast, mimicking the refined predictions at the cascade level using the prediction from the teacher proves to be effective for distillation.

Comparison with Baseline Models

Knowledge distillation helps boosting the generalization of smaller models. This is particularly true for MPM-2S which with distillation reaches the F-Score performance of its deeper (and slower) version MPM-4S baseline. There is still a small performance gap between the CPM-2S teacher and learner performance (0.03 in F-Score), but the later runs 10 times faster and is 101 times lighter. A visualization of the efficiency-accuracy is shown in Figure 4.8. We have marked MPM-2S trained with distillation with *. Interestingly, the same conclusions can be drawn from the distillation results obtained on the CMU-Panoptic dataset, with the MPM-2S improving by 0.1 its F performance using the same distillation approach.

4.4 Summary

This chapter has investigated different techniques to achieve fast 2D body landmark detection in multi-person pose estimation scenarios. Our work proposes to use depth images in combination with efficient CNNs to maintain the trade-off between speed and accuracy. We have introduced efficient instantiations of the pose machines architecture with stacked regressors and employing Residual, SqueezeNets and MobileNets designs.

We employed knowledge distillation to enhance the generalization capacities of our lightweight models. Our approach couples distillation with our cascade of detectors architectures performing distillation at each stage of the cascade. In a set of experiments, we have shown that leveraging on knowledge distillation we can boost the performance of our lightweight models, running at 112.3 FPS with the MobileNet-based architecture at a small performance loss.

In addition, we validated the use of synthetic depth images to cope with the lack of training data. Our study suggests that the fusion of sensor background data with synthetic depth images aids the models' generalization capabilities on real data.

5 Domain Adaptation for Pose Estimation

As we described in chapter 3, deep learning approaches require a vast amount of annotated data to achieve good generalization. In chapter 4 we showed that deep CNN models can obtain excellent results when training them with large amounts of synthetic data with large variability: human poses, shapes, multi-person settings and large amount of background scenarios. However, the good performance is only achieved provided a set of data for the target scenario with landmark annotations.

For developing algorithms for supervised learning, it is assumed that samples for training and testing follow the same probability distribution. In practice it is very costly to obtain large amounts of data with enough variability and with corresponding high quality annotations. State-of-the-art public databases such as (Lin et al., 2014) have used crowdsourcing tasks for labeling body landmarks in RGB images by human annotators. For the case of depth images, manual labeling of body landmarks is more laborious: the human body appears only as a blob making it more difficult to accurately localize landmarks in the image.

We have followed the approach of obtaining training data by synthesizing depth images. Although the cost to generate labeled data is low, the resulting images will hardly match the exact conditions of the target scenarios. As a consequence, training our models with synthetic data and testing in real scenarios will suffer a degradation in the testing performance since training and testing data distributions are different. This situation gives rise to the so-called covariate shift (Sugiyama and Kawanabe, 2012).

Domain adaptation (reviewed in Section 2.5) are a set of transfer learning techniques that aim to learn a predictor in the presence of the covariate shift. The topic has been largely explored, focusing mainly in the problem of RGB image classification (Tzeng et al., 2015; Ganin et al., 2016; Tzeng et al., 2017). In most approaches the visual domains mainly differ in the objects perspective, lighting, background, and objects are mostly image centered. These are not directly applicable in our scenario, i.e. landmark localization in depth images framed as a regression task.

This chapter, based on (Martínez-González et al., 2020a, 2018a), discusses domain adaptation techniques for multi-person pose estimation from depth images. We investigate different approaches.

- The first one, discussed in Section5.2, is to use unsupervised adversarial domain adaptation. It is a framework enabling learning a body landmark detector using only annotations on synthetic images while leveraging the information contained in large quantities of unannotated real images.
- A second approach is to fuse synthetic depth images of persons (for which the annotation is known) with background depth data from the real sensor, generating semi-synthetic data allowing the network to already learn sensor noise characteristics (discussed in Sections 3.1.4 and 5.3).
- We also explored inpainting methods in order to fill the spourious artifacts that arise from depth sensing in order to imitate the real images to look similar to those of the synthetic images (Section 5.3).

We contrast these techniques with a simple finetuning approach using a small set of annotated real images and highlight the limitations of the different approaches. We evaluate the methods in a dataset of real HRI scenarios of people interacting with the humanoid robot Pepper.

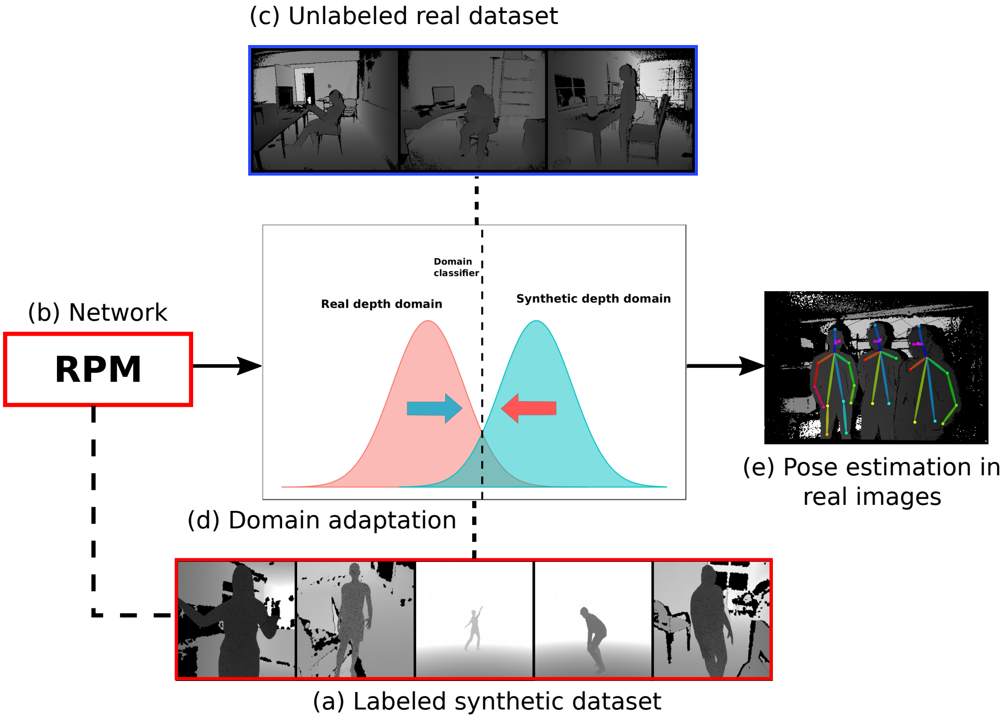


Figure 5.1 – Scheme for pose estimation learning and unsupervised domain adaptation. A CNN (b) is learned relying on synthetically generated images combining people under multiple pose configurations with varying real background images(a). The domain shift is addressed via an unsupervised adversarial domain adaptation method (d) that uses real unlabeled data (c).

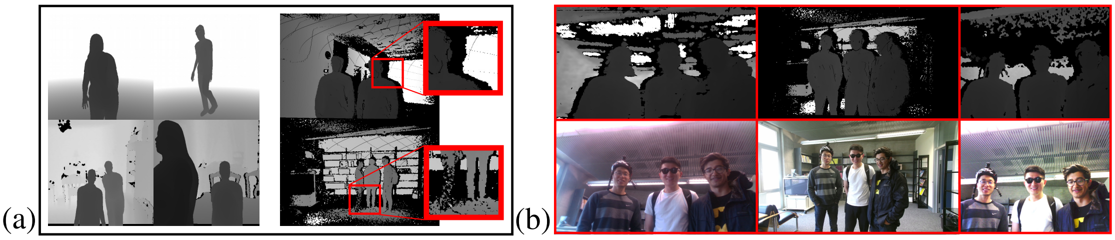


Figure 5.2 – Depth imaging characteristics of different sensors. (a) some visual characteristics of real depth images (right) like shadows around the silhouette or sensing failures due to surface material and depth variation (red square) are difficult to synthesize and therefore not present in the synthetic images (left); (b) HRI scene recorded with different RGB-D cameras. Left to right: Intel D435, Kinect 2 and Asus Xtion. Different depth sensors have different quality characteristics.

5.1 Depth Image Domains

Our training and testing setting is affected by the covariate shift issue. The problem arises from the differences in the visual features between synthetic and real depth images. On one side, training synthetic data (source domain) generated using rendering techniques are clean with smooth depth surfaces. On the other side, when testing with real depth images (target domain), the quality of the image greatly depends on the sensor specifications that affects the depth sensing (measurement variance, etc). Examples of these differences are shown in Figure 5.2. Some major ones are the shadows that appear around the silhouette of the person due to the triangulation process of depth sensing, and the missing values due to the reflectance properties of surfaces. These visual features are very difficult to simulate and therefore not present in the synthetic depth images

The performance gap between synthetic and real image domains can be alleviated provided that we are given enough annotated real depth images. Yet, manual annotation of body landmarks in depth images in large quantities is very time consuming. In this scenario, an unsupervised domain adaptation technique is desired to learn a predictor in presence of a domain shift.

5.2 Unsupervised Adversarial Domain Adaptation

An overview of this approach is illustrated in Figure 5.1. The unsupervised adversarial domain adaptation (ADA) can be stated as follows. We are given a source sample set (synthetic depth images) $\mathcal{S} = \{(\mathbf{I}_i^S, \mathbf{x}_i)\}_{i=1}^N$ and a target dataset (real depth images) $\mathcal{T} = \{\mathbf{I}_i^R\}_{i=1}^M$. Note that we are only given annotations of 2D keypoint locations $\mathbf{x}_i$ for the source samples $\mathbf{I}_i^S$. The goal is to learn a human pose predictor using the sample set $\mathcal{S}$ which performs well on data from $\mathcal{T}$, by mapping source and target data to an invariant representation. We follow the approach in (Ganin et al., 2016) in which more information can be found.

The distance between the source and target distributions of the input data can be measured via the H-divergence. As it is impractical to compute, it can be approximated by considering the generalization error of a domain classification problem. In essence, the distance will be minimum if a domain classifier is incapable of distinguishing between the samples from the different domains. Such domain confusion can be achieved by learning a mapping from the input data to an image representation invariant across domains.

To do so, we rely on the architecture presented in Figure 5.3(a). It is a multi-task architecture comprising three main components. The first one, G_f with parameters θ_f extracts features from the input image. Sharing these features, the branch G_y with parameters θ_y detects body landmarks and limbs in the image. In parallel the branch G_d with parameters θ_d classifies the input image into a domain label $d \in \{synthetic, real\}$. The adversarial adaptation procedure consists of learning a G_f network able to produce high level features sufficient for body landmark detection but which fool the domain classifier G_d.

More formally, let us denote by $\mathbf{F_I}$ the internal representation of image $\mathbf{I}$ in the network (features), which is computed as $\mathbf{F_I} = G_f(\mathbf{I}; \theta_f)$. We can then define as a measure of domain adaptation the opposite of the standard cross-entropy loss for the domain classifier

$$L_d(\theta_f, \theta_d) = -\frac{1}{N+M} \sum_{\mathbf{I} \in \mathcal{T} \cup \mathcal{S}} l_d(G_d(\mathbf{F_I}; \theta_d), y^d_{\mathbf{I}}), \tag{5.1}$$

where l_d is a logistic regression loss and $y^d_{\mathbf{I}}$ is the domain label associated with image $\mathbf{I}$. G_d is a domain classifier such that $G_d(\mathbf{F_I}; \theta_d) = 1$ if $\mathbf{I}$ is a real depth image and 0 otherwise. Optimizing the domain classifier is achieved by maximizing

$$R_d = \max_{\theta_d} L_d(\theta_f, \theta_d). \tag{5.2}$$

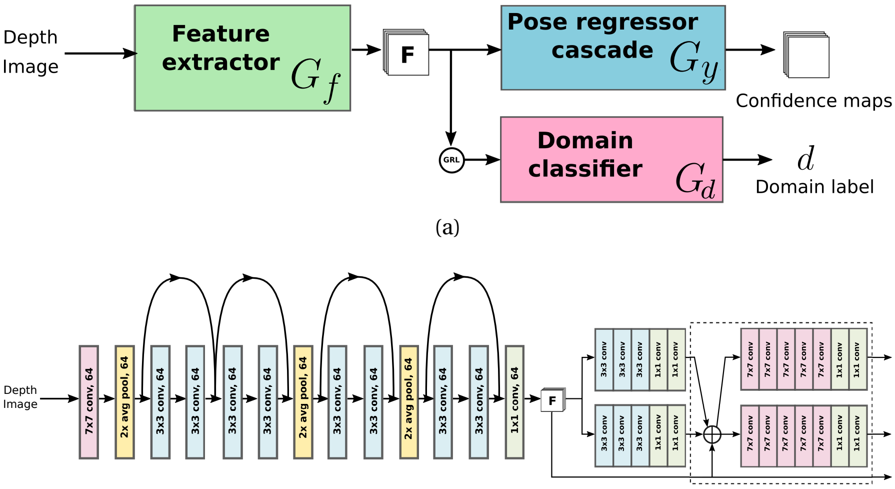
As shown in (Ganin et al., 2016), Eq (5.2) approximates the empirical H-divergence, measuring the similarity of the real and synthetic samples through the learned features $\mathbf{F_I}$. Such similarity measure can then be combined with the regression loss to define our adversarial loss:

$$L_{DA}(\theta_f, \theta_y, \theta_d) = L_{PM}(\theta_f, \theta_y) + \lambda \max_{\theta_d} L_d(\theta_f, \theta_d), \tag{5.3}$$

where λ represents the trade-off between landmark localization and domain adaptation, and $L_{PM}(\cdot)$ is defined by Eq. (5.6).

The adversarial learning process then consists of finding the optimal saddle point of the 'min-max' loss in Eq (5.3) by alternating the optimization of the parameters of the body landmark detector through the minimization $(\hat{\theta_f}, \hat{\theta_y}) = \arg\min_{\theta_f, \theta_y} L_{DA}(\cdot, \hat{\theta_d})$, and the maximization $\hat{\theta_d} = \arg\max_{\theta_d} L_{DA}(\hat{\theta_f}, \hat{\theta_y}, \theta_d)$. Therefore, θ_f evolves adversarially to increase the domain classification confusion while minimizing the error for landmark detection.

Due to the difference in nature between the pose regression and domain adaptation problems,



(b)

Figure 5.3 – CNN architectures for adversarial domain adaptation. The base domain adaptation architecture (a) is composed of a feature extractor module G_f and a pose regression cascade G_y implemented with the CNN architecture in (b) and it is extended with a domain classifier G_d for depth domain adaptation.

their losess involved in Eq (5.3) span different ranges. Therefore, the trade-off parameter λ has to reflect both the importance of the domain classification as well as to this difference between ranges. Its setting is detailed in Section 5.5.3.

5.3 Other Adaptation Methods

We consider other simpler adaptation methods based on data transformations during training and testing. Contrary to the unsupervised adversarial domain adaptation method, these do not necessitate of an auxiliary framework and can simply be used directly with the data at hand.

Semi-Synthetic Data. Data transformations such as fusing real depth background with synthetic silhouettes can help the model to adapt to sensor information, while learning with annotated synthetic data. Synthetic mages with fused real background, result in semi-synthetic data looking more realistic than plain synthetic images. Such a simple approach can improve the generalization of the model on real depth images given that sensor noise is included during learning.

We train our models with semi-synthetic data to add depth sensor information. During training, images are produced on the fly by randomly selecting one depth image background and body synthetic images, and generating a depth image using the character silhouette mask. We simply verified that there was a sufficient depth margin between the body foreground and the background, adding if necessary an adequate depth constant value to the entire background image. A more detailed description of the process is presented in Section 3.1.4.

Image Inpainting. Image inpatinting is a collection of techniques used for image restoration. Their principle is quite simple: replace "bad" pixel colors with its neighbouring pixels so that the replaced pixel looks like the neighborhood.

During testing we use image inpainting to make real depth images to look like synthetic ones and directly test with our models trained with semi-synthetic data. To do so, we treat as bad pixels those that typically correspond to missing values due to sensing errors and those in the depth shadows around objects silhouette caused by the depth sensing triangulation process. These pixels are localized by masking pixels with depth values equal to 0. We use the algorithm by (Telea, 2004) to inpaint these regions. The algorithm which works from the boundary of the bad regions to the inside, replacing the wrong values by a weighted sum of all the known pixel values in the neighborhood.

Finetuning. Practical CNNs models often have a large number of parameters. Train them from scratch with small dataset will result in overfitting. Finetuning a model (i.e. continuing training) previously trained on a large dataset but continuing a smaller different dataset, allows to use previously learned relevant features to perform in the new dataset. This can be seen as transfer learning provided the two related task are not drastically different.

We use finetuning on models trained with synthetic data to adapt them to real depth images. This is performed by simply continuing training the model with a small amount of labeled real depth image data as described in Section 5.5.3.

5.4 Network Architectures

5.4.1 Domain Classifier

We implemented the domain classifier G_d using a network composed of two average pooling layers with an intermediate layer of 1×1 convolution, and two fully connected layers before the classification sigmoid function at the end. We followed (Ganin et al., 2016) and included a gradient reversal layer (GRL) in the architecture to facilitate the joint optimization of Eq (5.3). The GRL acts as identity function during the forward pass of the network, but reverses the direction of the gradients from the domain classifier during backpropagation.

5.4.2 Pose Regressor CNN

We use the CNN architecture illustrated in Figure 5.3(b) for body landmark prediction. It is the Residual Pose Machines implementation of the convolutional pose machines architecture class introduced in Section 4.1.1.

Feature extraction network G_f. It consists of an initial convolutional layer followed by three residual modules with small kernel sizes (3 × 3). The network has three average pooling layers. Each residual module consists of two convolutional layers and a shortcut connection. Batch normalization and ReLU are included after all convolutional layers and shortcut connections as exemplified in Figure 4.3(a).

Pose regression cascade G_y. We maintain a large effective receptive field in the design of the branches $\phi_s(\cdot)$ and $\rho_s(\cdot)$. In the first prediction stage the network has three convolutional layers with filters of 3 × 3 and two layers with filters of 1 × 1, whereas in the remaining stages there are five and two convolutional layers with filters of 7 × 7 and 1 × 1 respectively.

5.4.3 Confidence Map Prediction and Pose Regression Loss

The CNN pose regressor is trained to predict confidence maps for the location of the different body parts and predict vector fields (part affinity fields) for the location and orientation of the body limbs.

The ideal representation of the body part confidence maps $\mathbf{H}^*$ encodes the ground truth location on the depth image as Gaussian peaks. The representation is modeled using Eq. (4.1). The ideal representation of the limbs $\mathbf{V}^*$ encodes the confidence for the connection between two adjacent body parts, in addition to information about the orientation of the limbs by means of a vector field. We use the vector field representation as in Eq. (4.2).

Training. Supervision is applied at the end of each prediction stage to prevent the network from vanishing gradients. This supervision is implemented by two L_2 loss functions, one for each of the two branches, between the predictions $\mathbf{H}_s$ and $\mathbf{V}_s$ and the ideal representations $\mathbf{H}^*$ and $\mathbf{V}^*$ for stage s

$$L_s^{\mathbf{V}} = \sum_{\mathbf{p}\in\mathbf{I}} ||\mathbf{H}_s(\mathbf{p}) - \mathbf{H}^*(\mathbf{p})||_2^2, \tag{5.4}$$

$$L_s^{\mathbf{H}} = \sum_{\mathbf{p}\in\mathbf{I}} ||\mathbf{V}_s(\mathbf{p}) - \mathbf{V}^*(\mathbf{p})||_2^2. \tag{5.5}$$

The final multi-task loss is computed as:

$$L_{PM} = \sum_{s=1}^{S} \left(L_s^{\mathbf{H}} + L_s^{\mathbf{V}}\right), \tag{5.6}$$

where S is the total number of prediction stages.

5.5 Experiments

5.5.1 Data

We considered the synthetic and real parts of our publicly available DIH dataset introduced in chapter 3. For both synthetic and real images, we performed data augmentation during training: rotation by a random angle within $[-20, 20]$ degrees with a 0.8 probability, and image cropping to the 368×368 training size with a probability of 0.9. The depth images are normalized by scaling linearly the depth values in the $[0, 8]$ meter range into the $[-0.5, 0.5]$ range.

Synthetic Data

We use the semi-synthetic data as the source domain in our experiments. The semi-synthetic data is generated during training as discussed in Section 5.3. Additionally we add pixel noise by randomly selecting 20% of the body silhouette's pixels and set their value to zero.

Real Data

In these experiments we use the unlabeled training set of real images comprising 6338 Kinect 2 depth images as the target domain [1]. For evaluation and comparison purposes we use the annotated folds as well which comprise 1750, 750 and 1000 images within the training, validation, and testing folds, respectively.

5.5.2 Evaluation Metrics

Pose Estimation Performance. We use standard precision and recall measures derived from the Percentage of Correct Keypoints (PCK) evaluation protocol as performance metrics, described previously in Section 4.3.2. Average recall (AR) and average precision (AP) values used to report performance are computed by averaging the landmark detection recall and precision values over landmark type and over several distance thresholds d to the ground truth.

5.5.3 Implementation Details

Pytorch is used in all our experiments. We train the Residual Pose Machine (RPM) network architecture with stochastic gradient descent with the momentum set to 0.9, the decay constant to 5×10^{-4}, and a batch of size 10. We initialized the learning rate to a value of 4×10^{-10} and decreased by a factor of 10 when a validation loss has settled. We limit the experiments to an

---

[1] Note that since ADA requires no annotation, it could be possible to collect and use much more images than the 6338 images, with the expectation of obtaining better results. This was not done here and left as future work.

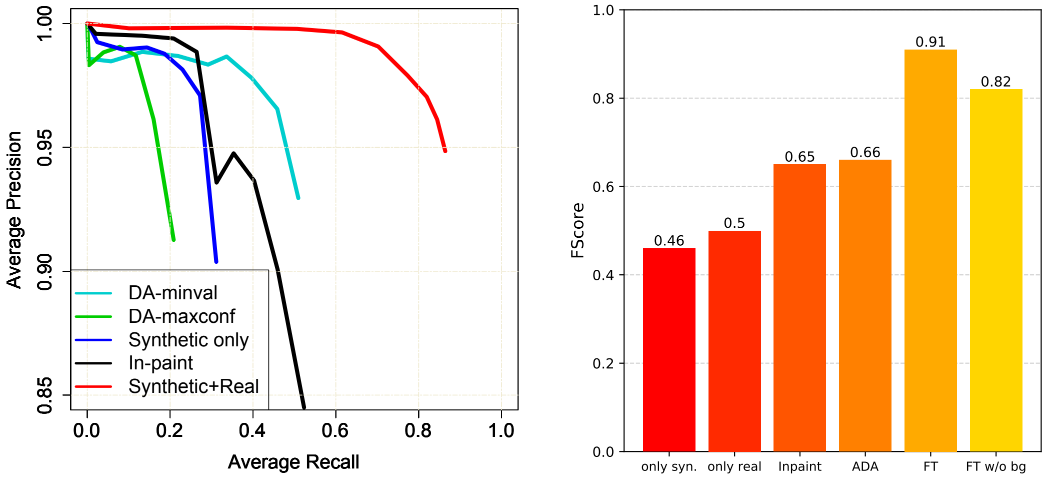


Figure 5.4 – Comparison of different techniques for synthetic-to-real domain adaptation.

Adaptation method	CR	AP	AR	F-Score
ADA - Min Loss	0.20	**92.95**	**50.94**	**0.66**
ADA - Max Confusion	0.50	91.26	20.86	0.40
Only synthetic	0.04*	90.37	31.14	0.46
Inpainting preprocessing	0.03*	84.49	52.33	0.65
Finetuning	0.15*	**93.96**	**87.72**	**0.91**
Finetuning (no background)	-	92.58	73.46	0.82

Table 5.1 – Comparison of performance obtained with the different techniques for domain adaptation using the RPM-2S model. CR stands for the confusion rate obtained in the domain classification task. Values with * were obtained by training a domain classifier using the corresponding model feature extractor as fixed features. Inpainting preprocessing corresponds to the application of an inpainting step on the real depth image before applying the detector.

architecture comprised of 2 prediction stages, i.e. RPM-2S.

Adversarial Domain Adaptation Training. We first train the feature extractor and cascade of predictors of the RPM-2S model with the semi-synthetic data for $200K$. Adversarial domain adaptation (ADA) is then performed by jointly training the domain classifier, feature extractor and cascade of predictors for another $100K$ iterations. Following common practices we gradually updated the trade-off parameter λ of eq (5.3) according to the training progress as $\lambda_p = \frac{2\Lambda}{1+exp(-10p)} - \Lambda$, where $p = t/T$, with t the current iteration and $T = 100K$. We set $\Lambda = 100$ so that the two losses in Eq (5.3) are in the same range.

Network Finetune Training. To finetune models we first train the feature extractor and cascade of predictors with the semi-synthetic data for $200K$ (13 epochs). Then train with real images for 100 epochs.

5.5.4 Results

Adaptation Criteria

During adaptation we monitored validation losses for body landmark detection and domain classification. After T iterations we selected domain adapted models according to either of two criteria on a validation set:

1. the minimum of a pose validation loss, and
2. maximum confusion (sum of false positives and false negatives for the domain classification task).

Table 5.1 reports the results where we also include the performance of models trained without adaptation.

The adversarial domain adaptation (ADA) framework aims at learning invariant features across domains. Nevertheless, we note that when maximum confusion is achieved the body landmark detection task is greatly hampered, and the model performs even worse than non-adapted models. In contrast, the model selected using a validation error on body landmark detection outperforms the non-adapted models while still achieving some level of domain confusion. We can notice that it is the recall performance measure which is greatly affected when changing the selected criteria. Certainly, a major difference between synthetic and real images is the lack of data around external edges that form the limbs extremities and silhouette. These are also the places where pose information is available. Confusing the domain classifier means that features exploiting this lack of data are removed, which hurts the body landmark detection performance.

Inpainting and Finetuning Methods

To understand the limits of our unsupervised ADA approach, we compare it with the performance of inpainting and finetuned models. Figure 5.4 shows the precision-recall curves of the different methods, while Table 5.1 summarizes these curves with the maximum F-Score obtained (also shown graphically in Figure 5.4 right).

We see that ADA slightly outperforms the inpainting approach. Indeed, the latter aims to fill the missing depth information as an image preprocessing step for non-adapted models. Observe however that inpainting greatly reduces precision, introducing artifacts in the image that are later confused as body landmarks or limbs. On the other hand, finetuning directly the network initially trained on synthetic images with even a small amount of labeled data greatly improves its generalization capabilities. Interestingly, note that without the addition of real background (removing the semi-synthetic approach), finetuned models perform worse given that the initial state of the network is not optimal arising from a representation learned with pure synthetic data.

5.6 Summary

In this chapter, we discussed domain adaptation approaches to train deep CNN for multi-person pose estimation in the presence of the covariate shift caused by training with synthetic images and testing in real ones.

We explored the unsupervised adversarial domain adaptation framework for our pose regression setting. Unsupervised adversarial domain adaptation aims to find domain-invariant features that fools a domain classifier in order to improve performance in the target domain without the use of annotated data. However, as shown by our experiments, achieving domain confusion might compromise the discriminant power of the learned features and leads to poor performance, in contrast to seek for the minimum in a pose validation loss.

We have compared the performance of unsupervised adversarial domain adaptation with three other approaches: 1) semi-synthetic data transformation approach that forms training examples with synthetic person silhouettes and real depth image backgrounds, 2) an inpainting method that makes a real noise image to look similar as a synthetic one, and 3) a simple finetuning method that exploits scarse labeled real data. Overall, our experiments suggest that domain adaptation solely improves the performance over synthetic-only trained models, and can be further boosted by finetuning on a small sample of labeled images when using pretraining with semi-synthetic data.

6 Multi-Person 3D Pose Estimation

In the previous chapters we have studied deep learning methods for 2D human pose estimation from depth images. Nevertheless, in HRI settings obtaining the 2D pose is a stepping stone for 3D human pose prediction and 3D scene understanding. For instance, in social HRI, the ability to sense the 3D pose of humans provides to the robot the means to understand a person's 3D motion for activity recognition and evaluate their interaction engagement.

This chapter is based on (Martínez-González et al., 2020b) and discusses a new method for fast and accurate multi-person 3D pose estimation for HRI scenarios. Although 3D pose estimation has been a very important topic of research, factors like person self occlusions, pose variations, sensing conditions and low computational budget increase the challenge of deploying accurate, reliable and efficient 3D pose estimation systems.

As reviewed in Section 2.3, methods can be grouped into two main threads: fitting and learning methods. The former use CNNs to localize 2D body parts and fit a 3D pose model with anatomical constraints with an optimization model. Learning based methods take advantage of CNN architectures to directly regress the 3D locations of body landmarks. Despite their great success the latter methods usually work over image crops centered around the person and rely on expensive voxelized representations. Thus, in the multi-person pose estimation setting the model has to be executed for each person in the scene making them computationally expensive.

The work for 3D pose estimation presented in this chapter relies on an approach that decouples 2D and 3D pose estimation. In Section 6.2.1 we introduce the regression CNNs architectures that we use to efficiently solve the 2D multi-person setting. These networks follow the pose machine architecture class previously discussed in Section 4.1.1 and predict body landmark confidences and body limb vector fields. Section 6.2.2 describes how we exploit the depth camera parameters to lift 2D detected landmarks to the 3D space. Our 3D pose estimation approach follows the learning based methods and regress 3D coordinates of body landmarks from lifted 2D landmarks (discussed in Section 6.3). The proposed decoupled approach is better suitable for the multi-person 3D pose estimation scenario which we show through a

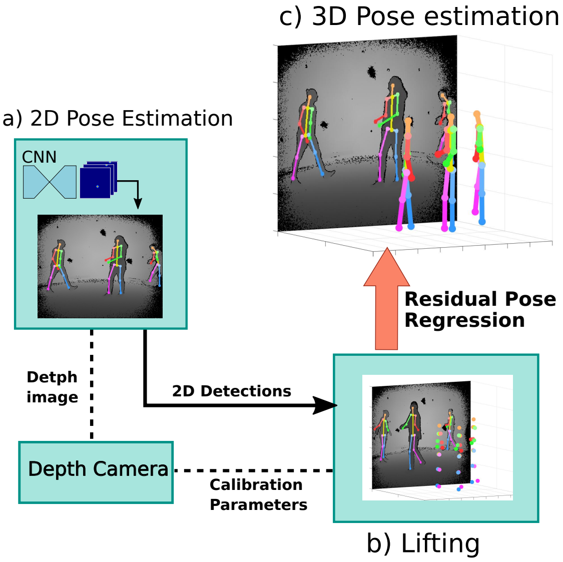


Figure 6.1 – Overview of our decoupled residual pose approach: a) bottom-up multi-person 2D pose detection; b) for each detected person, 2D body joints are lifted to the 3D space. c) 3D pose estimation using a residual pose regression network.

series of experiments described in Section 6.4.

6.1 Approach Overview

An overview of our approach for accurate and fast multi-person 3D pose estimation is presented in Figure 6.1. Our main idea is to better exploit the depth information and decouple the task in two main steps: 2D multi-person pose estimation and 3D pose regression.

Our motivations are that the first step can benefit from recent accurate and efficient architectures to achieve this task, and that the second one can be done efficiently by directly regressing the 3D pose coordinates from the 2D ones in two substeps: a simple but effective scheme which lifts the 2D estimates to 3D using the depth information and pose priors (to handle partial occlusion); and a novel efficient residual pose 3D regression method that works on this set of points. This makes our approach computationally lighter for multi-person HRI settings since compared to CNNs applied to image crops for 3D pose prediction, the cost of our 3D regression scheme is much smaller, and the cost saving is proportional to the number of people in the scene.

Specifically, the contributions of our work on 3D pose estimation can be summarized as:

- we investigate an innovative method decoupling the 3D pose estimation task into an accurate and efficient CNN-based 2D bottom-up multi-person pose estimation method and 3D pose regression;
- we propose a simple 2D-to-3D lifting scheme which handles 2D body joint miss detec-

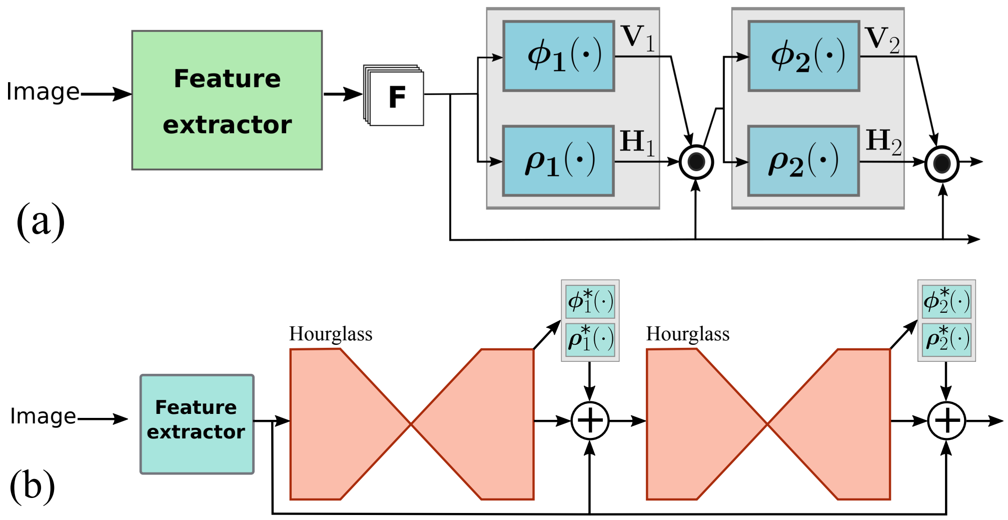


Figure 6.2 – CNN architectures used for 2D pose estimation. (a) Pose Machine architecture implemented by RPM and MPM (Martínez-González et al., 2020a). (b) Our extension of the Hourglass network for multi-person 2D pose estimation.

tions;

- we introduce a novel method for 3D pose regression from lifted 2D estimates by relying on a residual-pose deep-learning architecture;
- we demonstrate that despite its simplicity, our approach achieves very competitive results on different public datasets and is suitable for multi-party HRI scenarios.

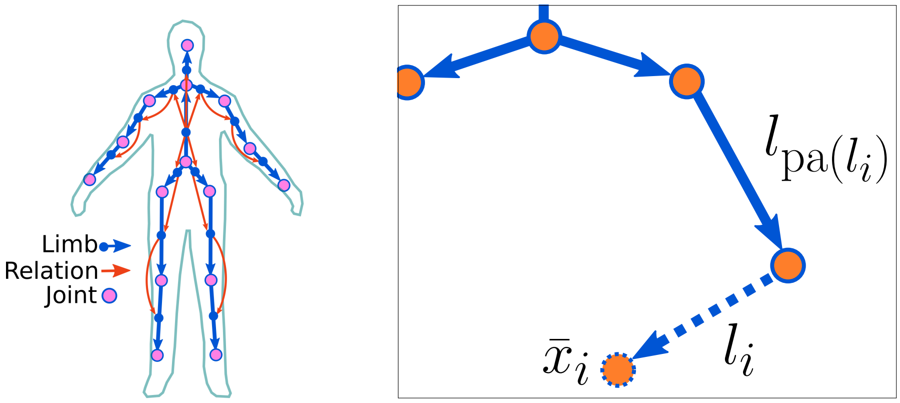
6.2 Efficient 2D Pose Estimation and Lifting

This section describes the CNN architectures used for accurate bottom-up 2D pose estimation and our proposed method for 2D-to-3D body joint lifting and for handling miss-detections due to (self-)occlusion or failures.

6.2.1 CNN-based 2D Pose Estimation

We follow recent breakthroughs in multi-person 2D pose estimation that use a CNN to predict confidence maps $\rho(\cdot)$ for the location of the body landmarks in the image and part affinity fields $\phi(\cdot)$ for the location and orientation of the limbs (Cao et al., 2017). We analyze different CNN architectures and the impact of their 2D estimates on the quality of the 3D pose.

Three architectures are considered. The two firsts are the efficient pose machines based on residual modules (RPM) and the one based on MobileNets (MPM) introduced in (Martínez-González et al., 2020a). These are lightweight CNNs that refine predictions with a series of prediction stages and are designed for efficient 2D pose estimation with real-time performance,



(a) (b)

Figure 6.3 – (a) Skeleton and limb pairwise relationships; (b) Process to recovery from 2D detection failures. The missing landmark $\bar{x}_i$ is estimated with the conditional of its limb $p(l_i | l_{\mathbf{pa}(l_i)})$.

see Fig.6.2 (a). Additionally, we consider the Hourglass network architecture (Newell et al., 2016) which was originally proposed for single person pose estimation. It comprises a series of UNet-like networks that process image features at different semantic levels. We follow the original design but adapt the output to predict part affinity fields to match our multi-person scenario by branching a duplicate of the confidence maps prediction layers (Fig.6.2 (b)).

6.2.2 Pose Lifting

Given 2D landmark detections, we use their corresponding depth values Z to lift them according to $\bar{x} = Z \cdot K \cdot (x_{img}, y_{img}, 1)^\top$, where $K = diag(1/f_x, 1/f_y, 1)$ is the depth camera matrix. However, different errors can arise. For example, a 2D detection might have missing depth value due to sensing failures. Additionally, as is common in typical HRI scenarios, self and between-person occlusions will naturally result in missing body detections.

In these cases, rather than feeding our regressor with dummy values which might bias estimations, we propose a simple recovery method. First, in case of missing depth values, we use the mean depth of the points with valid depth information in the landmark's vicinity. Second, in case of missed landmark detections, we rely on a 3D pose prior to infer their expected coordinates. However, rather than relying on expensive-to-compute prior (Sigal et al., 2011), we follow a simpler 3D limb prior based on pairwise relationships between limb vectors. Following a tree of limbs from the skeleton and taking the spine limb as root (see Fig. 6.4(a)), we consider adjacent limbs, encode their 3D direction and length within a joint Gaussian distribution $p(l_i, l_{\mathbf{pa}(l_i)})$, and learn the model parameters from training data. Then, to predict the lifted coordinates $\bar{x}_i$ of a missed landmark, we consider its associated limb l_i in the skeleton whose other landmark is already lifted, and compute the mean of the conditional Gaussian distribution $p(l_i | l_{\mathbf{pa}(l_i)})$ of l_i conditioned on its limb parent $\mathbf{pa}(l_i)$ to further compute

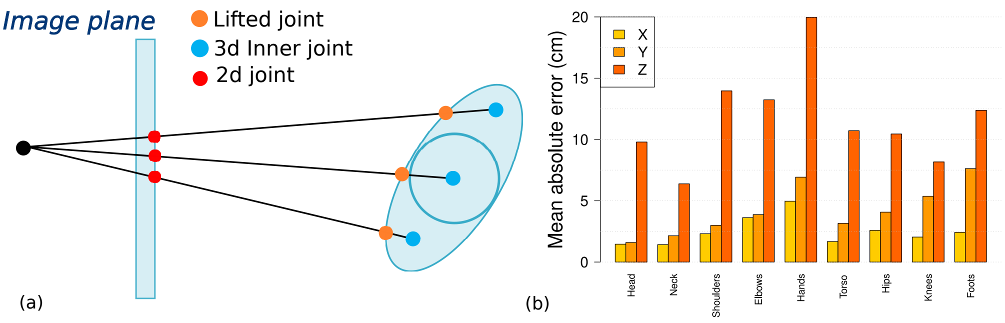


Figure 6.4 – (a) Illustration of the error introduced by the lifting process of the 2D detected landmarks; (b) mean absolute error on each coordinate when using the 3D lifted points as the 3D estimation on the ITOP dataset.

$\bar{x}_i$.

Note that our approach requires some body landmarks to be detected. Indeed, as in our opinion it is unrealistically to attempt determining the complete 3D pose of the person from a few detected body landmarks, e.g. the arm, we assume that at least the spine limb and other two body landmarks in the trunk (shoulders, heaps) are detected.

6.3 Human 3D Pose Estimation

This section presents our residual-pose learning approach to predict (in a camera coordinate frame) the 3D coordinates of a human skeleton comprising J body landmarks.

6.3.1 Residual Pose Learning

Provided the 2D body landmark detections, our lifting step provides a *rough* estimate of the 3D pose. Yet, lifted values will exhibit 3D pose estimation errors, specially since lifted 3D points lie on the depth surface rather than represent the inner joint (see Fig. 6.4). In this regard, in absence of other sources of errors (missed detections, occlusion, etc.) we can argue that such estimates differ only from the true 3D pose by some coordinate offset. This inspired us to follow a simple yet effective approach to obtain refined estimates from rough lifted estimates.

Our approach can be set as follows: given a *rough* 3D pose estimate $\bar{\mathbf{x}} \in \mathbb{R}^{J\times 3}$ obtained from the 2D landmark detection lifting step, and its *true* corresponding 3D pose $\mathbf{x}^* \in \mathbb{R}^{J\times 3}$, the neural regressor f can focus on modelling their residual $\mathbf{x}^* - \bar{\mathbf{x}}$ as:

$$f(\bar{\mathbf{x}}) + \bar{\mathbf{x}} = \mathbf{x}^*. \tag{6.1}$$

The function $f(\bar{\mathbf{x}})$ is the residual to be learned. Graphically, these residuals represent the vector of coordinate offsets that are necessary to predict the true 3D pose $\mathbf{x}^*$ (hence a residual pose). Architecturally speaking, the operation $f(\bar{\mathbf{x}}) + \bar{\mathbf{x}}$ is performed by a shortcut connection

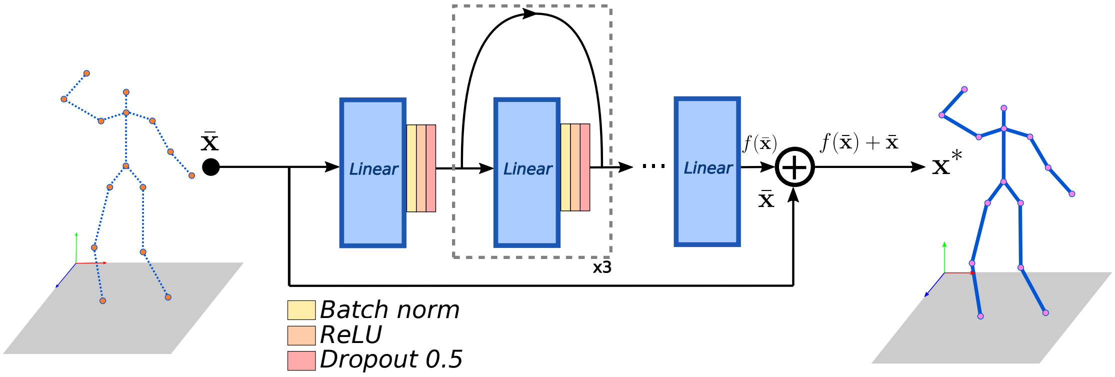


Figure 6.5 – Residual pose learning framework. Our neural network regressor receives as input a lifted 3D pose $\bar{\mathbf{x}}$. Due to the global skip connection, the regressor has to predict the residual pose $f(\bar{\mathbf{x}})$ to be added to $\bar{\mathbf{x}}$ to predict the true 3D pose. The building block of our neural network regressor is a linear layer followed by batch normalization, ReLU activations and dropout, and with a skip connection.

with the identity mapping of $\bar{\mathbf{x}}$, as shown in Fig. 6.5.

Additionally, we can augment $\bar{\mathbf{x}}$ by incorporating the confidence of the 2D detections provided by the 2D pose estimation CNN. This will add an extra dimension for each detected landmark $\bar{\mathbf{x}} \in \mathbb{R}^{J\times 4}$. In such case the shortcut connection works as a pooling layer that removes the extra dimension to match the one of $\mathbf{x}^*$. We analyze this particular case in Section 6.4.

6.3.2 Neural Network Regressor

We aim to find a simple and efficient network architecture f that performs well enough in the regression task. Fig.6.5 shows a diagram with the basic building blocks of our architecture. It is a multi-layer network consisting on a series of fully-connected layers, each followed by batch normalization, ReLU activations and dropout layers. The first layer receives as input the lifted pose $\bar{\mathbf{x}}$ and outputs 2048 features. This number of features are kept fixed until the output layer that generates the residual pose vector in $\mathbb{R}^{J\times 3}$. Each of the inner layers have skip connections. One can normally squeeze as many inner layers S to make the regressor deeper. However, we set $S = 3$.

6.3.3 Pose Learning Loss

Let $\hat{\mathbf{x}} = f(\bar{\mathbf{x}}) + \bar{\mathbf{x}}$ be the 3D pose prediction. We use the following loss to train our neural network regressor

$$L_{res} = \frac{1}{J}\sum_{i=1}^{J} ||\hat{\mathbf{x}}_i - \mathbf{x}_i^*||_1, \tag{6.2}$$

where $\mathbf{x}_i^*$ is the ground truth of the body landmark i and $\hat{\mathbf{x}}_i$ is the 3D prediction for such landmark. In our experiments we use the smooth L1 norm as we found out that it works better than the L2 or plain L1 norms.

6.4 Experiments

We conducted several experiments to evaluate our approach effectiveness in single and multi-person scenarios.

6.4.1 Depth Image Datasets

ITOP (Haque et al., 2016). This dataset consists of images in a single person pose estimation setting. It has 18k and 5k depth images for training and testing, respectively, recorded with an Asus Xtion camera. It was built from 20 subjects performing 15 different actions each. Section 3.2.3 provides more details of this dataset.

CMU-Panoptic (Joo et al., 2017). This dataset was previously introduced in Section 3.2.2. It comprises multiple recordings acquired with different sensor devices such as color and depth cameras (Kinect2). We consider a subset of the depth recordings from the *Haggling* category. The setup contains several interacting people with diverse body pose configurations with respect to the camera and between-person interactions. For training we selected 15k 3D person instances from the sequence 170407_haggling_a3 for training. For testing 1.5k 3D person instances were selected from the sequence 170407_haggling_b3.

6.4.2 Evaluation Metrics

Mean Average Precision (mAP). As standard practice in 3D human pose estimation, we use mean average precision at 10 cm (mAP@10cm) to measure the 3D detection performance. A successful detection is considered when the detected 3D body landmark falls within a distance less than 10 cm from the ground truth. We report the average precision (AP) for individual body landmarks and to measure the overall performance, the mean average precision (mAP) defined as the mean of the APs of all body landmarks. Larger values are better.

Mean Per Joint Position Error (MPJPE). It measures the average error in Euclidean distance between the detected 3D body landmarks and the ground truth. Lower values are better. We report MPJPE in centimeters for each body landmark and their mean for the overall performance (mMPJPE).

Percentage of Correct Keypoints (PCK). We use PCK to evaluate the performance of the 2D pose estimation task. It relies in the precision and recall that result from the percentage of correct detected keypoints (body landmarks). We follow the evaluation protocol presented in (Martínez-González et al., 2018b). For each joint (e.g. knee), true positives, false positives, and false negatives are counted using a radius obtained according to the height of the bounding

CNN model	MPM	RPM	HG
FPS	84	35	18
# Params	304.9K	2.84M	12.9M
F-Score (2D)	0.96	0.96	**0.97**
mAP@10cm	85.61	85.96	**85.97**
mMPJPE	6.83	7.18	**6.78**

Table 6.1 – 2D and 3D pose estimation performance obtained for the different 2D CNN architectures and their computational complexity.

box (ground truth) containing the person. Then, the precision and recall rates are calculated by averaging the above values over a set of varying radius, body landmarks, and dataset samples.

6.4.3 Implementation Details

Image Pre-Processing. We normalize the depth images by linearly scaling the depth sensor values in $[0,8]$ meter range into the $[-0.5,0.5]$ range.

2D CNN architectures and Training. We keep the performance-efficiency trade-off reported in (Martínez-González et al., 2020a) and experiment with RPM with 2 stages and MPM with 4 stages. We configure the Hourglass architecture (HG) to 2 stages as it was shown that performance saturates at this point (Newell et al., 2016).

We train the 2D pose estimation CNNs using Adam. To avoid overfiting due to the low number of depth images in the addressed datasets, and increase the 2D pose performance, we train the networks for 13 epochs with the large synthetic people dataset introduced in (Martínez-González et al., 2018b). Then, the CNNs are finetuned using the real dataset (ITOP or CMU-Panoptic) for 100 epochs.

Residual Pose Regressor. We train our neural network regressor for 200 epochs using Adam and minibatches of size 128. We apply standard normalization to the 3D lifted pose and the 3D ground truth pose by substracting the mean and dividing by the standard deviation. We select $1e-3$ as initial learning rate and decrease it by 2 every 20 epochs.

6.4.4 Experimental Results

2D Pose Network Architectures. We evaluate the quality of the 2D pose predictions for the 3D pose estimation task in the ITOP dataset. Fig. 6.6 shows the 2D pose estimation performance curves and the 3D pose error in terms of MPJPE for the different CNNs. Table 6.1 summarizes these results with the maximum F-Score obtained for 2D pose estimation, and the mAP and mMPJPE for 3D pose prediction. Indeed, providing better 2D pose estimates reflects directly in the 3D performance. Overall the HG 2D detections provide the best 3D estimates achieving

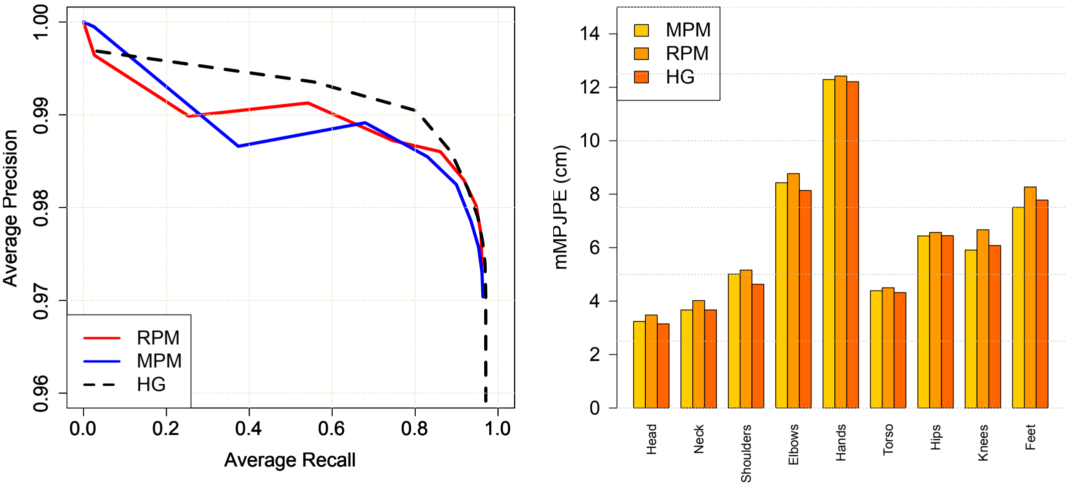


Figure 6.6 – Performance of the different CNNs for 2D pose estimation. Left: 2D pose estimation performance measured with recall and precision curves. Right: resulting 3D estimation pose performance in terms of MPJPE for each body part. The lower the better.

the lowest mMPJPE and better mAP. We select the HG network for the rest of the analysis.

Computational Requirements. Table 6.1 reports the number of parameters of each CNN and the frames per second (FPS) required for the forward pass in a single Nvidia card GTX 1050. Note that the FPS is also valid for the multi-person case since the CNNs predict the pose for each individual in the image in a single forward pass. Additionally, the neural network regressor requires 12.7M parameters but runs at 1700 FPS, so its cost, even when applied for multiple person, is negligible compared to that of a 2D pose CNN. Hence our proposed approach can run very efficiently in real-time in a single GPU.

Comparison with The State-Of-the-Art. Table 6.2 compares the detailed AP scores for each body landmark of our proposed approach (**R-Pose**) with the state of the art in the ITOP dataset. Overall our residual pose learning approach shows very competitive results obtaining the second best performance. The best performing work is (Moon et al., 2018) that processes voxelized representations of the 3D space processed with a 3D CNN, and uses an ensemble of 10 models for the final prediction. Contrary, our residual pose approach is simpler and efficient, requiring a single low budget GPU and achieving real time performance. Example results are shown in Fig. 6.7 (top row).

Multi-person 3D pose estimation. Table 6.2 reports AP and MPJPE for the multi-person setting in the CMU-Panoptic dataset. Naturally the ranges of pose profile, multiple scales, and the quality of sensing make this setup more challenging than the single person pose setting. The more affected body landmarks are the hands and elbows with lower AP and larger MPJPE. Note these are the elements that are in constant motion and are more affected by self

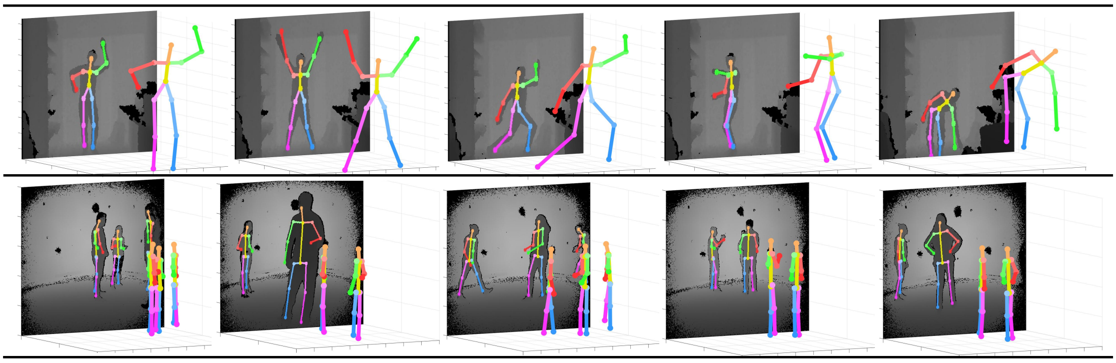


Figure 6.7 – 3D pose estimation examples and their 2D projection of our approach on the single person ITOP dataset (top row) and the multi-person CMU-Panoptic dataset (bottom row).

occlusions, compared to other elements like the torso and head. Fig. 6.7 shows prediction examples.

Recovery from 2D Failures. We report the results of removing the prior recovery component introduced in Section 6.2.2. Table 6.2 shows the performance for the single and multi-person settings (**R-Pose**$^{-}$). The performance drops especially for the multi-person scenario.

2D Landmark Detection Confidence. We incorporated the confidence of the 2D detections provided by the CNN that range in the $[0,1]$ interval by padding the confidences to the input lifted skeleton and proceed with our residual pose learning setting. When a landmark was recovered by the process discussed in Section 6.2.2, we set a low confidence value of $\sigma = 0.1$ to identify them from the rest. The results are reported in Table 6.2 (**R-Pose***). The mAP slightly decreases in this case. However, in the multi-person setting some especific elements (head, elbows, hands) have slightly better detection rate.

Coordinate Regression. We experimented with 3D coordinate regression using the neural network architecture introduced in Section 6.3.2 and predict X, Y, Z coordinates of the body landmarks from lifted 2D detections, dropping the residual pose connection. Table 6.2 compare these results (C-Reg) with our residual pose approach. The performance drops for both single and multi-person settings. Certainly when people appear roughly in the same position, as it is the case in ITOP dataset, 3D coordinate regression presents a good alternative. However, our residual pose approach outperforms the direct 3D coordinate regression in both, the single and multi-person settings.

6.5 Summary

In this chapter we addressed the problem of multi-person 3D pose estimation from depth images. We have introduced our proposed approach that decouples 3D pose estimation into

	ITOP (front-view)							
	AP@10cm							
Body part	(Jung et al., 2015)	(Haque et al., 2016)	(Guo et al., 2017)	(Moon et al., 2018)	**R-Pose**	**R-Pose***	**R-Pose**$^-$	C-Reg
Head	97.8	98.1	**98.7**	98.29	98.27	98.13	98.33	97.8
Neck	95.8	97.5	**99.4**	99.07	98.6	98.56	98.5	98.66
Shoulders	94.1	96.5	96.1	**97.18**	95.34	95.2	92.78	95.64
Elbows	77.9	73.3	74.7	**80.42**	76.52	75.89	74.38	74.24
Hands	**70.5**	68.7	55.2	67.26	61.69	61.28	59.98	55.01
Torso	93.8	85.6	98.7	**98.73**	98.56	98.64	98.62	97.57
Hips	80.3	72	91.8	**93.23**	90.07	90.31	89.4	87.09
Knees	68.8	69	89	**91.80**	89.13	88.93	88.82	88.29
Feet	68.4	60.8	81.1	**87.6**	84.28	83.52	83.66	83.99
Mean (mAP)	80.5	77.4	84.9	**88.74**	85.97	85.71	84.9	84.17

	CMU-Panoptic							
	MPJPE (cm)				AP@10cm			
Body part	**R-Pose**	**R-Pose***	**R-Pose**$^-$	C-Reg	**R-Pose**	**R-Pose***	**R-Pose**$^-$	C-Reg
Head	**6.59**	6.78	10.17	11.17	96.4	**96.67**	79.47	72.33
Neck	**7.29**	7.45	8.5	11.68	**96.53**	96.2	92.13	74.07
Shoulders	**8.55**	8.66	10.96	14.38	**87.17**	85.6	77.17	54.33
Elbows	14.52	**14.19**	23.86	20.2	59.17	**61.97**	38.3	28.93
Hands	**27.85**	27.96	31.16	26.37	16.63	17.47	**17.77**	6.37
Torso	9.06	**8.51**	9.92	11.93	**93.27**	92.67	87.6	67.53
Hips	**8.57**	8.67	12.16	12.99	**91.97**	90.27	70.1	66.1
Knees	**9.24**	9.43	14.72	13.96	**81.8**	80.6	58.67	52.33
Feet	11.26	**11.19**	18.8	15.54	**70.77**	70.5	52.17	48.27
Mean	**12.2**	**12.2**	16.79	16.11	**73.41**	73.22	59.17	48.44

Table 6.2 – Visualization of 3D pose estimation performance. Top: mAP of the state-of-the-art on single person pose estimation setting in the ITOP dataset. Bottom: mAP and mMPJPE for the multi-person pose estimation setting in the CMU-Panoptic dataset.

2D pose estimation and 3D pose regression applied on lifted detections. The benefits of decoupling is that we can efficiently solve 2D pose with lightweight CNN architectures and 3D pose can be estimated efficiently from the 2D estimations. We have investigated different CNN architectures for 2D pose estimation all of which are based in the pose machine architecture class and introduced a pairwise 3D limb prior to recover from 2D detection failures. Our 3D pose regression approach works in a residual-pose regression learning to predict the 3D pose by refining lifted detections.

Despite the simplicity of our approach we achieve competitive results in two public datasets for single and multi-person pose estimation. Our method proposes a more efficient alternative for multi-party HRI settings than state-of-the-art methods that operate on single person centered image crops at a time. Our study opens the way for new research. One limitation of our model is that it does not consider the skeleton kinematics in the learning process. Additionally, body motion modelling can be introduced to introduce temporal consistency in our 3D predictions.

7 Human Motion Prediction

An important ability of an artificial system aiming at human behaviour understanding resides in its capacity to apprehend the human motion, including the possibility to anticipate motion and behaviour (e.g. reaching towards objects). This chapter is based on (Martínez-González et al., 2021) and introduces a deep learning-based method for human motion prediction from sequences of 3D human poses, poses which can be extracted with our approach introduced in chapter 6.

Human motion prediction has been a hot topic researched for decades. With the recent popularity of deep learning, Recurrent Neural Networks (RNN) have replaced conventional methods that relied on Markovian dynamics (Lehrmann et al., 2014) and smooth body motion (Sigal et al., 2011), and instead they learn these properties from data. However, motion prediction remains a challenging task due to the non-linear nature of the articulated body structure. Although the different motions of the body landmarks are highly correlated, these properties are hard to model in learning systems.

A family of RNN-based approaches have proposed to frame the task of human motion prediction as sequence-to-sequence problem. These methods usually rely on stacks of Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU) and solve the task with autoregressive decoding: generating predictions one at a time conditioned on previous predictions (Martinez et al., 2017a; Aksan et al., 2019). This practice has two major shortcomings. First, autoregressive models are prone to propagate prediction errors over time. Elements of the predicted sequence are conditioned to previous predictions (containing a degree of error) resulting in an increment of their error contents. Second, autoregressive modelling is not parallelizable which may cause deep models to be more computationally intensive since the elements in the predicted motion sequence are generated sequentially, one at time.

Since the breakthrough of the Transformer neural network in machine translation (Vaswani et al., 2017), it has been adopted in other research areas for different sequence-to-sequence tasks such as automatic speech recognition (Katharopoulos et al., 2020) and object detection (Carion et al., 2020). These methods leverage the long range memory of the attention

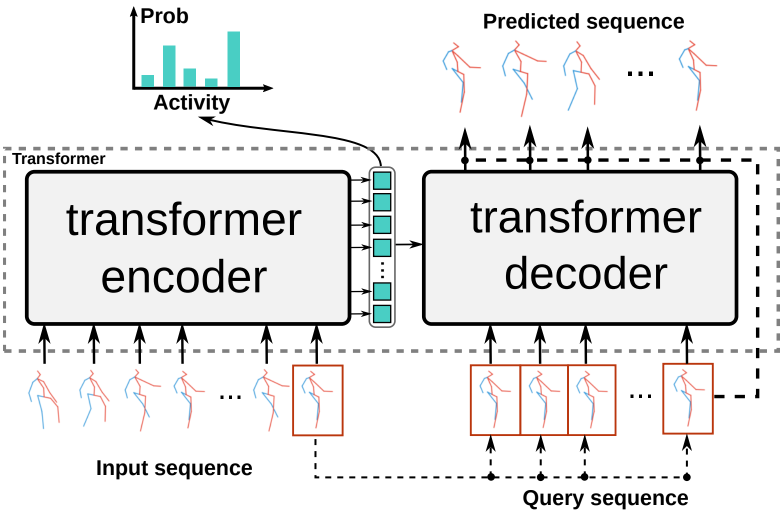


Figure 7.1 – Proposed motion prediction approach. A transformer with non-autoregressive decoding predicts future motion from a generated sequence composed from a query pose.

modules to identify specific entries in the input sequence which are relevant for prediction, a shortcoming of RNN models. During training, the Transformer allows parallelization with *look ahead* masking. Yet, at testing time, they use an autoregressive setting which makes it difficult to leverage the parallelization capabilities. Hence, autoregressive transformers exhibit large inference processing times hampering their use in applications that require real-time performance such as in HRI.

Our work for motion prediction presented in this chapter aims to reduce computation cost and potentially avoid error propagation with a non-autoregressive approach using the Transformer neural network. In Section 7.2.2 we present a non-autoregressive Transformer with self- and encoder-decoder attention. In Section 7.2.3 we introduce different approaches to compute pose embeddings from 3D skeletons (3D pose representation) and to predict motion sequences. Section 7.2.4 introduces our approach that uses the encoded motion sequence to perform human activity classification using the encoder self-attention embeddings. In Section 7.3 we present a series of experiments with the results of our approach for human motion prediction and skeleton-based activity classification. We show that our non-autoregressive Transformers obtain competitive results with state-of-the-art approaches. Finally, Section 7.4 describes our ongoing and future research work.

7.1 Approach Overview

An overview of our approach for non-autoregressive motion prediction is presented in Figure 7.1. We use a non-autoregressive transformer decoder that predicts the 3D skeletons simultaneously (in parallel) from a *query sequence* generated in advance rather than predicting one 3D skeleton at a time conditioned the previous predictions. As a consequence, the predicted sequence is generated in a single decoding pass resulting in a more efficient approach.

Our work is inline with recent sequence-to-sequence approaches using transformers with parallel decoding (Carion et al., 2020; Gu et al., 2018). Contrary to state-of-the-art methods that rely only in a transformer encoder for human motion prediction (Aksan et al., 2021; Wei et al., 2020), our approach uses as well a transformer decoder architecture with self- and encoder-decoder attention. Inspired by recent research in non-autoregressive machine translation (Gu et al., 2018), we generate the inputs to the decoder with elements from the input sequence.

In addition, we explore the inclusion of activity information by predicting as well activity from the input sequences. Modelling motion and activity prediction jointly has not often been investigated by previous works, though these topics are highly related. Hence, we propose a skeleton-based activity classification by classifying activities using the encoder self-attention predictions generated from the input motion sequence. We train our models jointly for activity classification and motion prediction and show the potential of this multi-task framework.

7.2 Method

The goal of our study is to explore solutions for human motion prediction leveraging the parallelism properties of transformers during inference. In the following sections we introduce our Pose Transformer (POTR), a non-autoregressive transformer for motion prediction and skeleton-based activity recognition.

7.2.1 Problem Formulation

Given a sequence $\mathbf{X} = \{\mathbf{x}_{1:T}\}$ of 3D poses we seek to predict the most likely immediate following sequence $\mathbf{Y} = \{\mathbf{y}_{1:M}\}$, where $\mathbf{x}_t, \mathbf{y}_t \in \mathbb{R}^N$ are N-dimensional pose vectors (skeletons). This problem is strongly related with conditional sequence modelling where the goal is to model the probabilities $P(\mathbf{Y}|\mathbf{X};\theta)$ with model parameters θ. In our work, θ are the parameters of a Transformer neural network.

Given its temporal nature, motion prediction has been widely addressed as an autoregressive approach in an encoder-decoder configuration: the encoder takes the conditioning motion sequence $\mathbf{x}_{1:T}$ and computes a representation $\mathbf{z}_{1:T}$. The decoder then generates pose vectors $\mathbf{y}_t$ one by one taking $\mathbf{z}_{1:T}$ and its previous generated vectors $\mathbf{y}_{\tau<t}$. While this autoregressive approach explicitly models the temporal dependencies of the predicted sequence $\mathbf{y}_{1:M}$, it requires to execute the decoder M times. This becomes computationally expensive for very large transformers, which in principle have the property of parallelization (exploited during training). Moreover, autoregressive modelling is prone to propagate errors to future predictions: predicting pose vector $\mathbf{y}_t$ relies in predictions $\mathbf{y}_{\tau<t}$ which in practice contain a degree of error. We address these limitations by modelling the problem in a non-autoregressive fashion

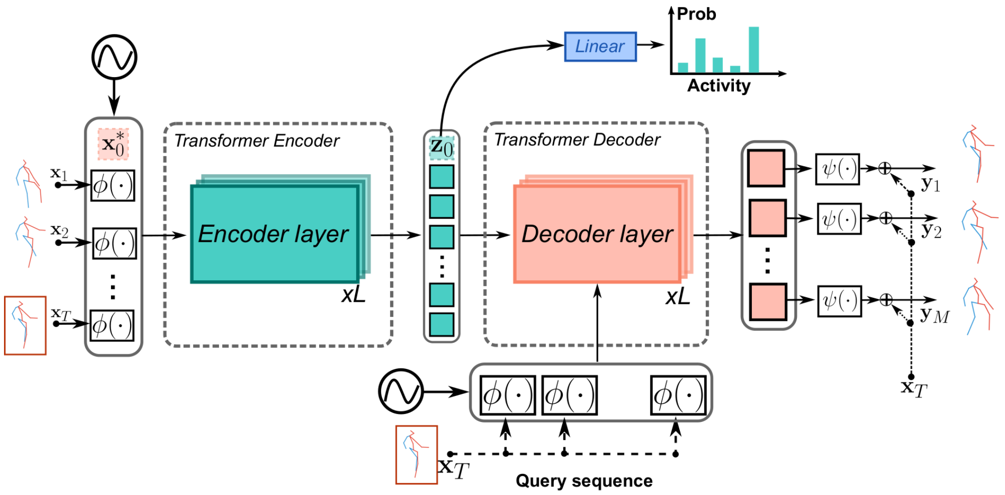


Figure 7.2 – Overview of our approach for non-autoregressive human motion prediction. Our model is composed of networks ϕ and ψ, and a non-autoregressive Transformer built on feed forward networks and multi-head attention layers as in (Vaswani et al., 2017). First, a network ϕ computes embeddings for each pose in the input sequence. Then, the transformer encoder generates self-attention maps from the input sequence of embeddings. Finally, the predicted sequence is generated with network ψ in a residual fashion. Activity classification is performed by adding a learnable *class token* $\mathbf{x}_0$ to the input sequence.

as we describe in the following.

7.2.2 Pose Transformers

The overall architecture of our POTR approach is shown in Figure 7.2. Similarly to the original Transformer (Vaswani et al., 2017), our encoder and decoder modules are composed of feed forward networks and multi-head attention modules. While the encoder architecture stays unchanged from the original encoder model, the decoder works in a non-autoregressive fashion to avoid error accumulation and reduce computational cost.

Our POTR comprises three main components: a pose encoding neural network ϕ that computes pose embeddings for each 3D pose vector in the input sequence, a non-autoregressive transformer, and a pose decoding neural network ψ that computes a sequence of 3D pose vectors. While the transformer learns the temporal dependencies, the functions ϕ and ψ shall identify spatial dependencies between the different body parts for encoding and decoding pose vector sequences.

More specifically, our architecture works as follows. First, the pose encoding network ϕ computes an embedding of dimension D for each pose vector in the input sequence $\mathbf{x}_{1:T}$. The transformer encoder takes the sequence of pose embeddings (agreggated with positional embeddings) and computes the representation $\mathbf{z}_{1:T}$ with a stack of L multi-head self-attention layers. The transformer decoder takes the encoder outputs $\mathbf{z}_{1:T}$ as well as a query sequence

$\mathbf{q}_{1:M}$ and computes an output embedding with a stack of L multi-head self- and encoder-decoder attention layers. Finally, pose predictions are generated in parallel by the network ψ from the decoder outputs and a residual connection with the query sequence. We detail each component in the following.

Transformer Encoder. It is composed of L layers. Figure 7.3 illustrates the layer architecture. Each layer has a standard architecture consisting of multi-head self-attention modules and a feed forward networks (point-wise). The encoder receives as input the sequence of pose embeddings of dimension D added with positional encodings and produces a sequence of embeddings $\mathbf{z}_{1:T}$ of the same dimensionality.

Transformer Decoder. Our transformer decoder follows the standard architecture: it comprise L layers of multi-head self- and cross-attention modules and feed forward networks (see Figure 7.3). In our work, every layer in the decoder generates predictions. The decoder receives a query sequence $\mathbf{q}_{1:M}$ and encoder outputs $\mathbf{z}_{1:T}$ and produces M output embeddings in a single pass. These are then decoded by the network ψ into 3D body skeletons.

The decoding process starts by generating the input to the decoder $\mathbf{q}_{1:M}$. As remarked in (Gu et al., 2018) given that non-autoregressive decoding exhibits complete conditional independence between predicted elements $\mathbf{y}_t$, the decoder inputs should account as much as possible for the time correlations between them. Additionally, $\mathbf{q}_{1:M}$ should be easily inferred. Inspired by non-autoregressive machine translation (Gu et al., 2018), we use a simple approach filling $\mathbf{q}_{1:M}$ using copied entries from the encoder inputs.

More precisely, each entry $\mathbf{q}_t$ is a copy of a selected *query pose* from the encoder inputs $\mathbf{x}_{1:T}$. We select the last element of the sequence $\mathbf{x}_T$ as the kernel pose and fill the query sequence with this entry. Given the residual learning setting, predicting motion can be seen as predicting the necessary pose offsets from last conditioning pose $\mathbf{x}_T$ to each element $\mathbf{y}_t$. We have found this strategy to work better than a uniform selection from the input sequence as proposed in (Gu et al., 2018) or pure learnable embeddings like in (Carion et al., 2020).

7.2.3 Pose Encoding and Decoding

Input and output sequences are processed from and to 3D pose vectors with networks ϕ and ψ respectively. The network ϕ is shared by the transformer encoder and decoder. It computes a representation of dimension D for each of the 3D skeletons in the input and query sequences. The decoding network ψ transforms the M decoder predictions of dimension D to 3D skeletons independently at every decoder layer.

The aim of the ϕ and ψ networks is to model the spatial relationships between the different elements of the body structure. To do this, we investigated two approaches. In the first one we consider a simple approach setting ϕ and ψ with simple single linear layers. In the second approach we follow (Wei et al., 2020) and use Graph Convolutional Networks (GCN) that

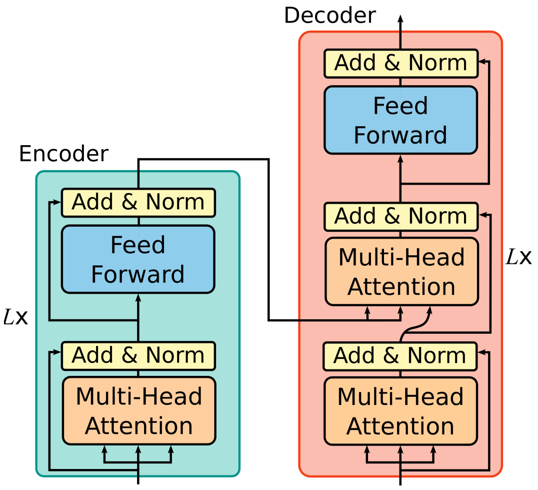


Figure 7.3 – Encoder and decoder architectures each with L layers. As in (Vaswani et al., 2017), each layer is composed of feed forward networks and multi-head attention layers.

densely learn the spatial connectivity between body parts.

To make our manuscript self contained, we briefly introduce how GCNs work in our human motion prediction approach. Given a feature representation of the human body with K nodes, a GCN learns the relationships between nodes with the strength of the graph edges represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$. Examples of representations are body skeletons or embeddings. A GCN layer l takes as input a matrix of node features $\mathbf{H}_{l-1} \in \mathbb{R}^{K \times F}$ with F features per node, and a set of learnable weights $\mathbf{W}_l \in \mathbb{R}^{F \times O}$. Then, the layer computes output features

$$\mathbf{H}_l = \sigma(\mathbf{A}_l \mathbf{H}_{l-1} \mathbf{W}_l), \tag{7.1}$$

where σ is an activation function. A network is composed by stacking layers which aggregates features of the vicinity of the nodes.

Our GCN architecture is shown in Figure 7.4. Adjacency matrices $\mathbf{A}_l$ and weights $\mathbf{W}_l$ are learnt. It is composed of graph convolution layers followed by batch normalization, *tanh* activations and dropout layers. The internal consist of S residual connections. The first layer takes as input a set of features $\mathbf{H}_0$, i.e. skeletons or decoder embeddings, and outputs $F = 512$ features per node. This dimension is kept fixed until the output layer that generates pose embeddings or skeleton sequences. Though we can normally squeeze as many inner layers, we set $S = 1$.

7.2.4 Activity Recognition

Activity can normally be understood as a sequence of motion of the different body parts in interaction with the scene context (objects or people). In our method, the transformer encoder

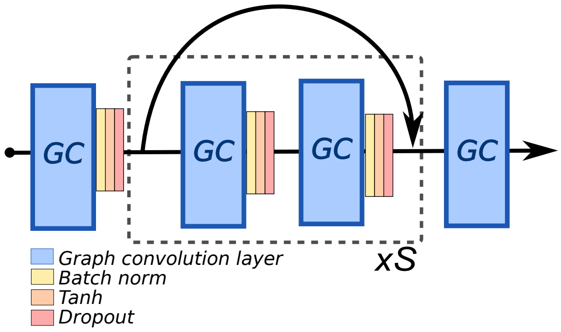


Figure 7.4 – Overview of our Graph Convolutional Network architecture. It is composed of graph convolition layers followed by *tanh* activations, batch normalization, and dropout layers. Similar to (Wei et al., 2020), our architecture comprises S residual connections. In practice we used $S = 1$.

encodes the body motion with a series of self-attention layers. We explore the use of encoder outputs $\mathbf{z}_{1:T}$ for activity classification (as a second task) and train a single linear layer classifier to determine the action corresponding to the motion sequence presented as input to the Transformer.

We explore two approaches. The first approach consist on using the entire transformer encoder outputs $\mathbf{z}_{1:T}$ as input to the classifier. However, these normally contain many zeroed entries suppressed by the probability maps normalization in the multi-head attention layers. Naively using these for activity classification might lead our classifier to struggle in discarding these many zero elements. Therefore, similar to (Dosovitskiy et al., 2020), we include a specialized *class token* in the input sequence to store information about the activity of the sequence. The class token $\mathbf{x}_0$ is a learnable embedding that is padded to input sequence to form $\mathbf{x}_{0:T}$. In the output of encoder embeddings $\mathbf{z}_{0:T}$, $\mathbf{z}_0$ works as the activity representation of the encoded motion sequence. To perform activity classification we feed $\mathbf{z}_0$ to a single linear layer to predict class probabilities for C activity classes (see Figure 7.2).

7.2.5 Training

We train our model in a multi-task fashion to jointly predict motion and activity. Let $\hat{\mathbf{y}}^{l}_{1:M}$ be the predicted sequence of N-dimensional pose vectors at layer l of the transformer decoder. We compute the layerwise loss

$$L_l = \frac{1}{M \cdot N} \sum_{t=1}^{M} ||\hat{\mathbf{y}}^{l}_{t} - \mathbf{y}^{*}_{t}||_1, \tag{7.2}$$

where $\mathbf{y}^{*}_{t}$ is the ground truth skeleton at target sequence entry t. The overall motion prediction loss is computed by averaging the losses of all decoder layers

$$L_{motion} = \frac{1}{L}\sum_{l=1}^{L} L_l. \tag{7.3}$$

We train our pose transformer with loss

$$L_{POTR} = L_{motion} + \lambda L_{activity}, \tag{7.4}$$

where $L_{activity}$ is the standard multi-class cross entropy loss.

7.3 Experiments

We conducted several experiments to evaluate our method for the tasks of motion prediction and activity recognition.

7.3.1 Data

Human 3.6M. We use the Human 3.6M dataset (Ionescu et al., 2014) in our experiments for human motion prediction. The dataset depicts seven actors performing 15 activities, e.g. walking, eating, sitting, etc. We follow standard protocols for training and testing (Martinez et al., 2017a; Aksan et al., 2019; Wei et al., 2019). Subject 5 is used for testing while the others for training. Input sequences are 2 seconds long and testing is performed over the first 400 ms of the predicted sequence. Evaluation is done in a total of 120 sequences across all activities by computing the angle error between predictions and ground truth in Euler angles representations.

NTU Action Dataset. The NTU-RGB+D (Shahroudy et al., 2016) dataset is one of the biggest and challenging benchmark datasets for human activity recognition. It is composed of 58K Kinect 2 videos of 40 different actors performing 60 different actions from different viewpoints. We follow the cross subject evaluation protocol provided by the authors that comprises of 40K sequences for training and 16.5K for testing. We use the 3D skeletons provided by the dataset for our motion prediction task and measure the performance using the mean average precision (mAP) at 10cm. Given the small length of the sequences, we feed our POTR with sequences of 1.3 seconds (40 frames) and predict sequences of 660 ms (20 frames) long.

7.3.2 Implementation details

Data Preprocessing. We apply standard normalization to the input and ground truth skeletons by substracting the mean and dividing by the standard deviation. For the H3.6M dataset we

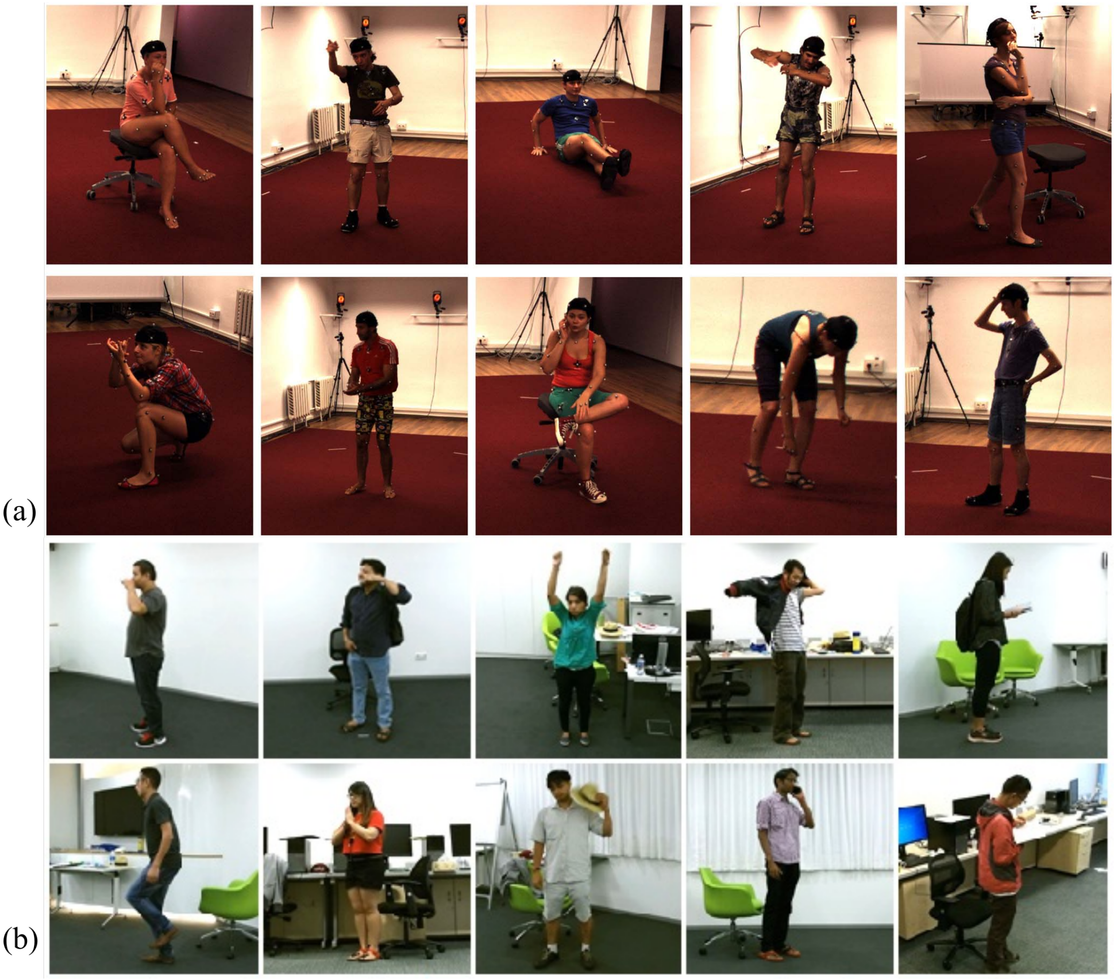


Figure 7.5 – Examples of images from: (a) H3.6M dataset, and (b) the NTU action dataset.

remove global translation of the skeletons and represent the skeletons with rotation matrices (then transformed to Euler angles for testing). Skeletons in the NTU dataset are represented in 3D coordinates and are centred by subtracting the spine joint.

Training. We use Pytorch as our deep learning framework in all our experiments. Our POTR is trained with AdamW (Loshchilov and Hutter, 2019) and setting the learning rate to 10^{-04} and weight decay to 10^{-05}. POTR models for the H3.6M dataset are trained during 100K steps with warmup schedule during 10K steps. For the NTU dataset we train POTR models during 300K steps with warmup schedule during 30K.

Models. We set the dimension of the embeddings in our POTR models to $D = 128$. The multi-head attention modules are set with pre-normalization and four attention heads. For the experiments with GCN architectures we set the number of inner layers to $S = 1$. We did not observe benefits with a larger number of inner residual layers.

7.3.3 Evaluation metrics

Euler Angle Error. We follow standard practices to measure the error of pose predictions in the H3.6M dataset by computing the error between the Euler angle skeleton representations. First, pose predictions are transformed from rotation matrices to Euler angles. Then the error is computed with the Euclidean norm between predictions and ground truth.

Mean Average Precision (mAP). We use mAP@10cm to measure the detection performance. A successful detection is considered when the predicted 3D body landmark falls within a distance less than 10 cm from the ground truth. The mean average precision (mAP) is defined as the mean of the APs of all body landmarks.

Mean Per Joint Position Error (MPJPE). MPJE measures the average error in Euclidean distance between the predicted 3D body landmarks and the ground truth.

7.3.4 Results

Evaluation on H3.6M Dataset

In this section we validate our proposed approach for motion prediction and compare it against the state-of-the-art using the H3.6M dataset.

Non-Autoregressive Prediction. We compare the performance of our non-autoregressive transformer with its autoregressive version. The autoregressive version does not use the query pose and predicts pose vectors one at a time from its own predictions fed to the decoder. Table 7.1 reports the obtained Euler angle errors. Our non-autoregressive approach shows lower error than its counter part (POTR-AR) in most of the time intervals.

milliseconds	80	160	320	400	560	1000
POTR-AR	0.23	0.57	0.99	1.14	1.37	1.81
POTR	0.23	**0.55**	**0.94**	1.08	1.32	1.79
POTR-GCN (enc)	**0.22**	0.56	**0.94**	**1.01**	**1.30**	**1.77**
POTR-GCN (dec)	0.24	0.57	0.96	1.10	1.33	1.77
POTR-GCN (full)	0.23	0.57	0.96	1.10	1.33	1.80

Table 7.1 – **H3.6M** prediction performance. Top: autoregressive (POTR-AR) and non-autoregressive POTR models using linear layers for networks ϕ and ψ. Bottom: non-autoregressive POTR models with GCNs for network ϕ (enc), network ψ (dec) and both (full). Values correspond to the Euler angle error performance averaged over all actions. Lower values are better.

Pose Encoding and Decoding. We experimented with the networks ϕ and ψ using either linear layers or GCNs. Table 7.1 reports the results. We indicate when models are trained with GCN in the encoder (enc), decoder (dec) or in both (full). We observe that the use of GCN reduces the errors when it is applied exclusively to the encoder. Using a shallow GCN ($S = 1$) $PoseDecFn$ might be a weak attempt to decode pose vectors. However, we observed that the small size of the H3.6M dataset might not be enough to learn deeper architectures.

	Walking				Eating				Smoking				Discussion			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero Velocity (Martinez et al., 2017a)	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	0.26	0.48	0.97	0.95	0.31	0.67	0.94	1.04
Seq2seq. (Martinez et al., 2017a)	0.28	0.49	0.72	0.81	0.23	0.39	0.62	0.76	0.33	0.61	1.05	1.15	0.31	0.68	1.01	1.09
AGED (Gui et al., 2018)	0.22	0.36	0.55	0.67	0.17	**0.28**	0.51	0.64	0.27	0.43	**0.82**	0.84	0.27	0.56	**0.76**	**0.83**
RNN-SPL (Aksan et al., 2019)	0.26	0.40	0.67	0.78	0.21	0.34	0.55	0.69	0.26	0.48	0.96	0.94	0.30	0.66	0.95	1.05
DCT-GCN (ST) (Wei et al., 2020)	0.18	**0.31**	**0.49**	**0.56**	0.16	0.29	0.50	0.62	0.22	0.41	0.86	**0.80**	0.20	**0.51**	0.77	0.85
ST-Transformer (Aksan et al., 2021)	0.21	0.36	0.58	0.63	0.17	0.30	**0.49**	**0.60**	0.22	0.43	0.88	0.82	0.19	0.52	0.79	0.88
POTR-GCN (enc)	**0.16**	0.40	0.62	0.73	**0.11**	0.29	0.53	0.68	**0.14**	**0.39**	0.84	0.82	**0.17**	0.56	0.85	0.96

Table 7.2 – **H3.6M** comparison with the state-of-the-art for the common *walking, eating, smoking* and *discussion* for across different horizons. Values are the euler angle error.

Comparison with the State-Of-The-Art. Tables 7.2 and 7.3 compares our method with the state-of-the-art in terms of angle error for all the activities in the dataset. Results obtained with linear layers and GCN for encoding sequences are shown. Our POTR often obtains the first and second lower errors in all the activities, specially for the first time ranges. We obtain the lowest average error in the range (80ms). The use of the last input sequence entry as the query pose helps significantly to reduce the error in the immediate ranges. However, this strategy introduces larger errors for longer horizons where the difference between further pose vectors in the sequence and the query pose is larger. These effects can be observed in Figure 7.6, where we show the ground truth and predictions sampled every two positions in the target sequence.

Attention Weights Visualization. In Figure 7.7(a) we visualize the encoder-decoder attention

Model	Directions				Greeting				Phoning				Posing			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Seq2seq (Martinez et al., 2017a)	0.26	0.47	0.72	0.84	0.75	1.17	1.74	1.83	0.23	0.43	0.69	0.82	0.36	0.71	1.22	1.48
AGED (Gui et al., 2018)	0.23	0.39	**0.63**	**0.69**	0.56	0.81	1.30	1.46	**0.19**	**0.34**	**0.50**	**0.68**	0.31	0.58	1.12	1.34
DCT-GCN (ST) (Wei et al., 2020)	0.26	0.45	0.71	0.79	0.36	**0.60**	**0.95**	**1.13**	0.53	1.02	1.35	1.48	0.19	**0.44**	**1.01**	**1.24**
ST-Transformer (Aksan et al., 2021)	0.25	**0.38**	0.75	0.86	0.35	0.61	1.10	1.32	0.53	1.04	1.41	1.54	0.61	0.68	1.05	1.28
POTR-GCN (enc)	**0.20**	0.45	0.79	0.91	**0.29**	0.69	1.17	1.30	0.50	1.10	1.50	1.65	**0.18**	0.52	1.18	1.47

	Purchases				Sitting				Sitting down				Taking photos			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Seq2seq. (Martinez et al., 2017a)	0.51	0.97	1.07	1.16	0.41	1.05	1.49	1.63	0.39	0.81	1.40	1.62	0.24	0.51	0.90	1.05
AGED (Gui et al., 2018)	0.46	0.78	**1.01**	**1.07**	0.41	0.76	1.05	1.19	0.33	0.62	0.98	1.10	0.23	0.48	0.81	0.95
DCT-GCN (ST) (Wei et al., 2020)	0.43	0.65	1.05	1.13	0.29	**0.45**	**0.80**	**0.97**	0.30	**0.61**	**0.90**	**1.00**	0.14	**0.34**	**0.58**	**0.70**
ST-Transformer (Aksan et al., 2021)	0.43	0.77	1.30	1.37	0.29	0.46	0.84	1.01	0.32	0.66	0.98	1.10	0.15	0.38	0.64	0.75
POTR-GCN (enc)	**0.33**	**0.63**	1.04	1.09	**0.25**	0.47	0.92	1.09	**0.25**	0.63	1.00	1.12	**0.12**	0.41	0.71	0.86

	Waiting				Walking Dog				Walking Together				Average			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Seq2seq. (Martinez et al., 2017a)	0.28	0.53	1.02	1.14	0.56	0.91	1.26	1.40	0.31	0.58	0.87	0.91	0.36	0.67	1.02	1.15
AGED (Gui et al., 2018)	0.24	**0.50**	1.02	**1.13**	0.50	0.81	1.15	**1.27**	0.23	0.41	0.56	0.62	0.31	0.54	0.85	0.97
DCT-GCN (ST) (Wei et al., 2020)	0.23	**0.50**	**0.91**	1.14	0.46	0.79	**1.12**	1.29	**0.15**	**0.34**	**0.52**	**0.57**	0.27	**0.52**	**0.83**	**0.95**
ST-Transformer (Aksan et al., 2021)	0.22	0.51	0.98	1.22	0.43	**0.78**	1.15	1.30	0.17	0.37	0.58	0.62	0.30	0.55	0.90	1.02
POTR-GCN (enc)	**0.17**	0.56	1.14	1.37	**0.35**	0.79	1.21	1.33	**0.15**	0.44	0.63	0.70	**0.22**	0.56	0.94	1.01

Table 7.3 – Prediction results for the reminder of the 11 actions in the **H3.6M** dataset with our main non-autoregressive transformer.

maps for some of the activities in the dataset. Figure 7.7(b) shows the attention between elements of the input and predicted sequences for the *walking* action. The thickness of the line is proportional to the attention weight. Only weight values larger than the median of the attention map are shown. Note there exist high dependency between elements in the second half of the input sequence and first elements in the predicted sequence showing larger attention values. We also observe some elements in the predictions that attend to distant dependencies in the input sequence, specially when the overall body motion in both sequence is low e.g. *eating*.

Computational Requirements. We measured the computational requirements of models POTR and POTR-AR by the number of sequences per second (SPS) of their forward pass in a single Nvidia card GTX 1050. We tested models with 4 layers in encoder and decoder, and 4 heads in their attention layers. We input sequences of 50 elements and predict sequences of 25 elements. POTR runs at 149.2 SPS while POTR-AR runs at 8.9 SPS. Therefore, the non-autoregressive approach is less computationally intensive.

Evaluation on NTU Dataset

This section presents our results on motion prediction and activity recognition on the NTU dataset.

Motion Prediction Performance. Table 7.4 compares our POTR with the different decoding settings using the mAP metric (higher is better). Notice that removing the activity loss ($\lambda = 0$) slightly drops the performance for the longer horizons, indicating that predicting the activity

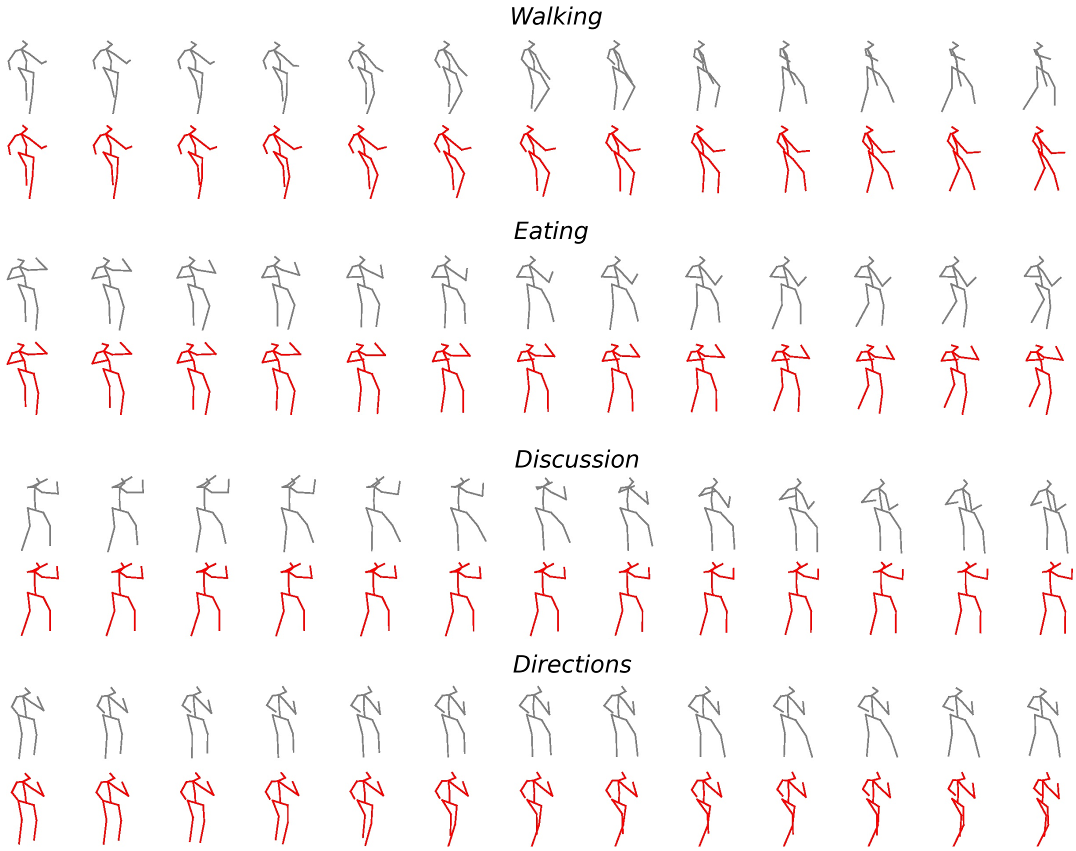
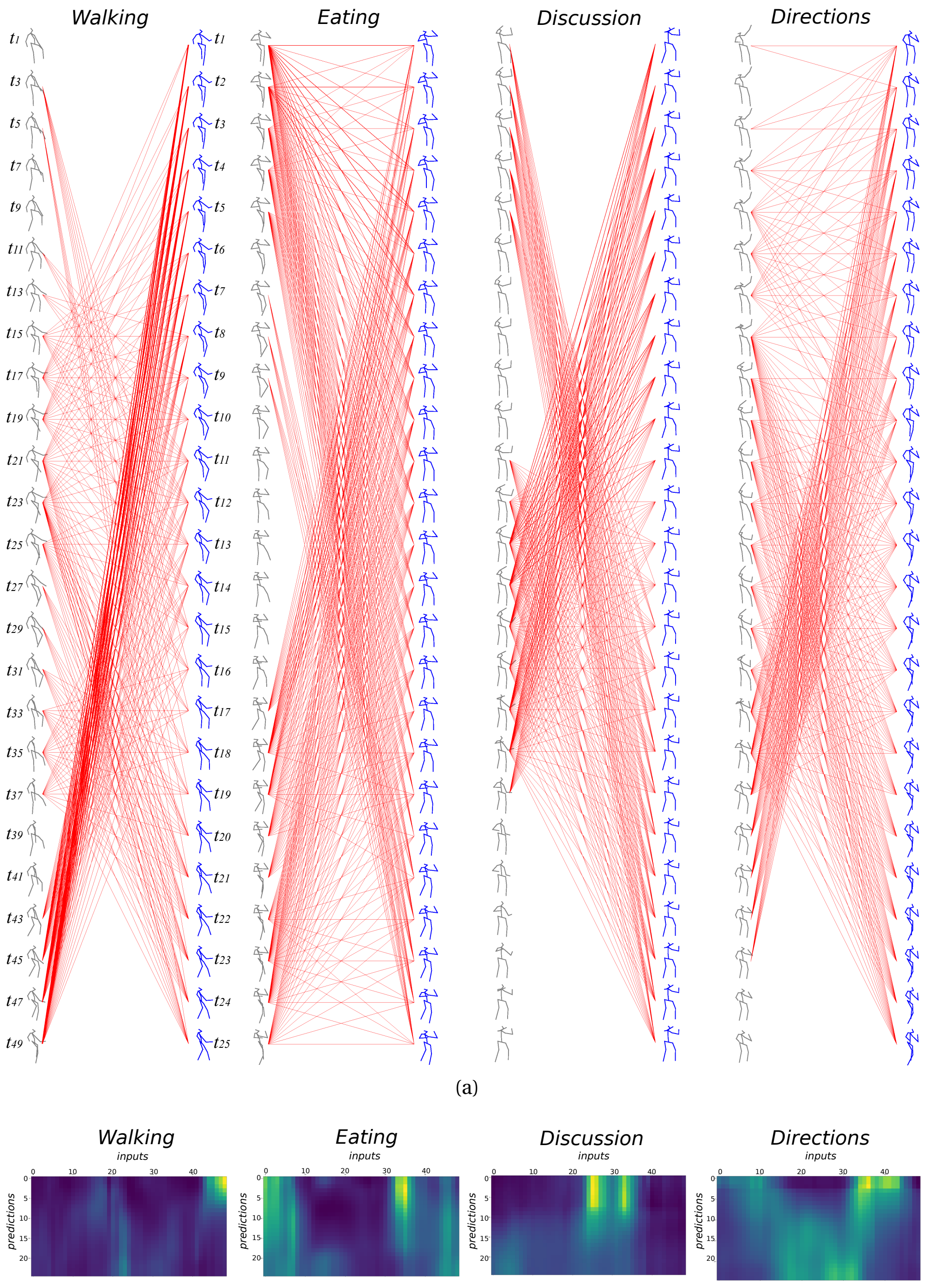


Figure 7.6 – Qualitative results for the H36M dataset. We show results for four actions and show ground truth and predicted elements coloured in gray and red respectively. Ground truth and predictions are sampled every two positions of the target sequence.

milliseconds	80	160	320	400	500	660	avg	accuracy
POTR-AR	0.96	0.92	0.85	0.83	0.80	0.76	0.76	0.32
POTR	0.96	**0.93**	**0.89**	**0.87**	**0.86**	**0.84**	**0.84**	**0.38**
POTR ($\lambda = 0$)	0.96	**0.93**	**0.89**	**0.87**	0.85	0.83	0.83	-
POTR (memory)	0.96	0.92	0.88	**0.87**	0.85	0.83	0.83	0.30
POTR-GCN (enc)	0.96	0.92	0.88	**0.87**	0.85	0.83	0.83	0.27
POTR-GCN (dec)	0.96	0.92	0.88	0.86	0.85	0.83	0.83	0.34
POTR-GCN (full)	0.95	0.90	0.85	0.84	0.82	0.79	0.79	0.30

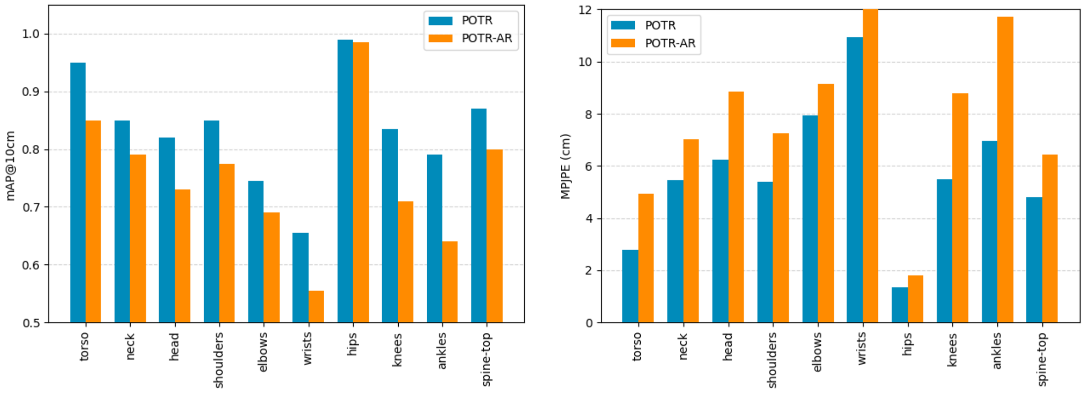
Table 7.4 – **NTU** motion prediction performance on different time ranges with autoregressive and non-autoregresive POTR. Model marked with *memory* replace the class token with the encoded memory for activity classification. Values correspond to the mean average precision in a 10 cm range (mAP@10cm). Higher values are better.



(a)

(b)

Figure 7.7 – **H3.6M** datasest encoder-decoder attention weight visualization. (a) Attention weights between input (gray) and predicted (blue) skeleton sequences. Only weights larger than the median are visualized. The thickness of the lines are proportional to the attention weights. For visualization purposes we show only half of the input sequence; (b) Raw encoder-decoder attention maps. Input and predicted entries are represented by columns and rows respectively.



(a) (b)

Figure 7.8 – NTU per body part motion prediction performance in terms of (a) mAP@10cm (the higher the better) and, (b) MPJPE (the lower the better).

class contributes to improve the motion prediction. The non-autoregressive setting shows higher mAP than the autoregressive setting. This is specially visible for longer time horizons. However, notice that the use of GCN to define networks ϕ and ψ does not bring many benefits compared to using linear layers, specially in the *full* case.

Figure 7.8 compares their overall per joint mAP and MPJPE using linear layers for ϕ and ψ. We removed results for the root-joint as we obtained zero MPJPE and perfect mAP scores. As observed in the plots, autoregressive setting shows larger MPJPE and lower mAP than the non-autoregressive case, specially for the body extremities (arms and legs components).

Activity Recognition. We test the performance of activity classification using a specialized class token associated to activity classes. Table 7.4 compares the different POTR configurations using the classification accuracy. Using a specialized token outperforms using the complete sequence of encoder embeddings $\mathbf{z}_{1:T}$ (marked as *memory*). Clearly, given that the self-attention embeddings might contain many non-informative zeroed values the classifier could get trapped in an attempt to ignore them. We obtain the best performance when using the specialized token with ϕ and ψ as linear layers. Figure 7.9 provides more classification details with the activity confusion matrix.

Table 7.5 compares our overall accuracy with state-of-the-art methods performing activity classification from sequences of 3D skeletons or color images. The first revelation is that our approach performs inline with state-of-the-art methods with the lowest accuracy. This is the case for methods that use only skeletal information. Among this category, the method presented by (Shahroudy et al., 2016) achieves the largest accuracy. They use a stack of LSTM modules with specialized part-based cells grouping them into arms, torso and legs. Instead, our approach computes memory for the entire body motion. The best performance overall is obtained by (Luvizon et al., 2018) which combines color image and skeleton modalities to

Method	Skeletons	RGB	Accuracy
Skeletal quads (Evangelidis et al., 2014)	√	-	38.62 %
2 Layer P-LSTM (Shahroudy et al., 2016)	√	-	62.93 %
Multi-task (Luvizon et al., 2018)	√	√	**85.5** %
Multi-task (Luvizon et al., 2018)	-	√	84.6 %
Ours POTR	√	-	38.0 %
Ours POTR (memory)	√	-	30.0 %

Table 7.5 – Activity classification performance comparison with the state-of-the-art in the **NTU** dataset using different input modalities.

classify activity. By all means including context of the activity provides extra information that cannot be extracted from working only with skeletal data such as certain objects of interaction.

7.4 Summary

In this chapter we addressed the problem of human 3D motion prediction from a sequence of 3D poses. We have introduced our pose transformer method (POTR), a non-autoregressive transformer for motion prediction. The obtained benefits are, first, that it does not propagate the error to long term horizons by conditioning on spurious predictions as in autoregressive decoding. Secondly, as it produces estimation in parallel it is more efficient at testing time than the autoregressive setting. Finally, we have leveraged in the encoder self-attention embeddings to perform activity classification. Despite its simplicity we have obtained competitive results on motion prediction in the H3.6M dataset.

Our work opens the door for more research. One of the main drawbacks is the error at long term in motion sequences that arises by feeding the decoder with copies of a single query pose. A better strategy is to rely in a set of query poses selected, for example, using the encoder self-attention embeddings by position modelling such as in (Gu et al., 2018). At the same time activity classification could benefit from using a transformer encoder that works over image crops of people and use a class token for classification. Final classification could be performed by aggregating the image based class token and the skeleton based token.

The transformer neural network with large breakthroughs such as Bert (Devlin et al., 2019) or GPT2 (Radford et al., 2019) first rely on a large dataset to pre-train the architecture and subsequently finetune on the target data. This practice could also benefit our pose transformer by, for instance, relying a large synthetic dataset such as (Mahmood et al., 2019) for pre-training and subsequently finetune on our smaller datasets.

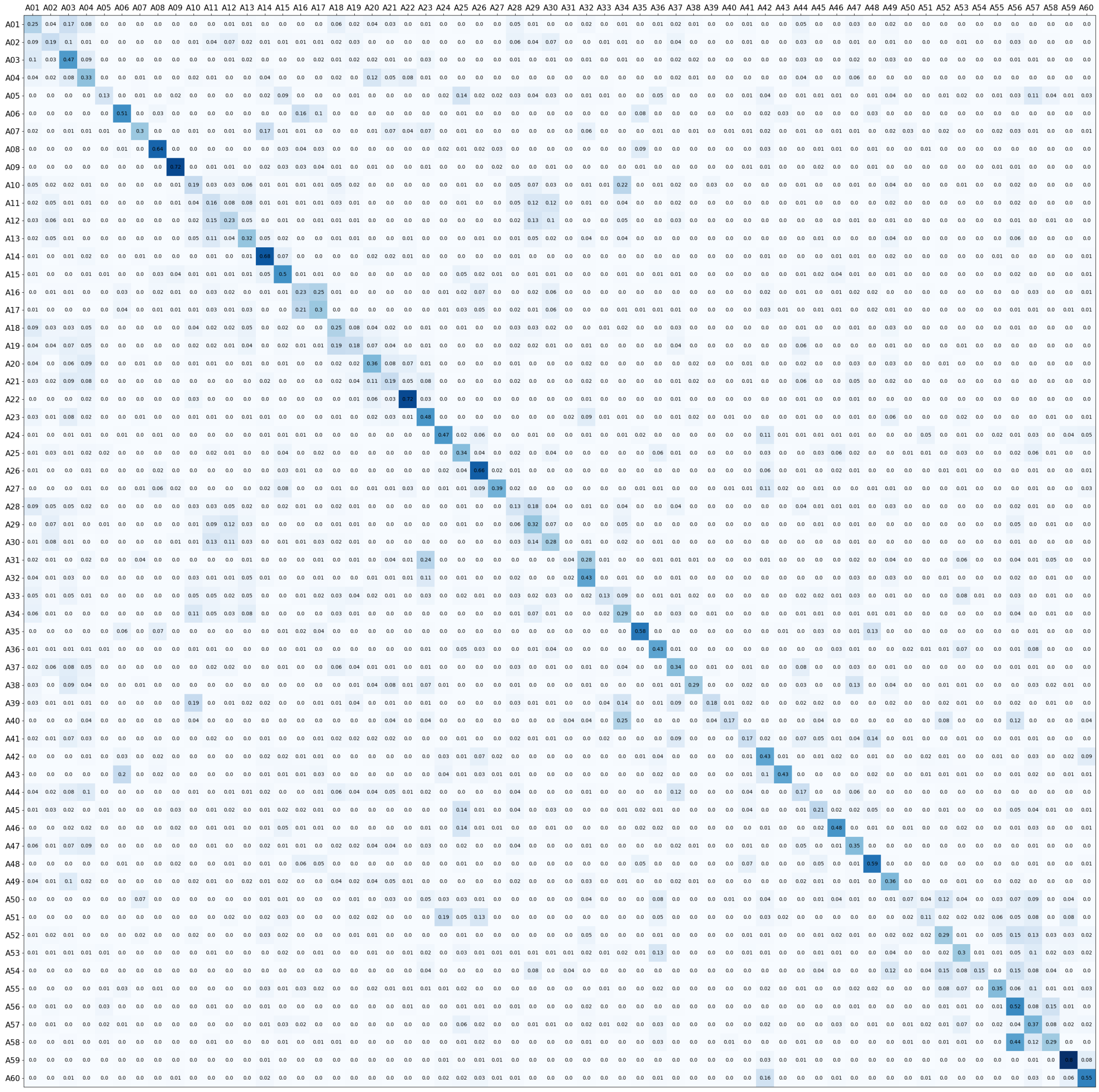


Figure 7.9 – Confusion matrix for the NTURGB+D dataset.

8 Conclusions and Future Directions

8.1 Summary of Contributions

Throughout this dissertation we have investigated efficient deep learning techniques for different components of the human-behaviour understanding pipeline. More concretely, this thesis covered the following aspects

- Generation of a large scale synthetic depth image database with annotations;
- The design of efficient CNN 2D and 3D pose estimation architectures;
- The study of domain adaptation techniques for adopting DNN pose estimation models trained with synthetic data and adapt to the real data;
- Non-autoregressive human motion prediction relying on transformers.

Our contributions have focused on the design of efficient DNN models and the exploitation of the depth image information to perform real-time 2D and 3D pose estimation and motion prediction in multi-person HRI applications.

Current deep learning approaches for 2D and 3D pose estimation are usually difficult to use in practical applications due to their large computational cost. Our approaches presented in chapter 4 addressed this limitation by relying on lightweight and efficient CNN architectures that perform inference in real-time. We have composed our lightweight CNN with Residual, MobileNet and SqueezeNet modules and pre-train them with synthetic depth images. All our models are implementation of the pose machine architecture class that incorporates a cascade of detectors to refine predictions.

We have shown that using synthetic data with both two-person instances and real background fusion largely improve the generalization capacities of our models after adapting them to real depth images. We have proposed to couple our cascade of detectors architectures with knowledge distillation to further improve the efficiency of our lightweight models. Our experiments showed that our models achieve good accuracy-speed trade-off and are inline in performance with large and accurate state-of-the-art CNN-based methods.

The success of deep learning methods rely on very large corpus with high quality annotations for supervised learning in order to avoid overfitting. We have addressed the need of training data for our target HRI scenario using synthetic depth images generated by computer graphics tools, avoiding the burden of data collection and manual annotation. Since training models with synthetic images and testing with real ones gives rise to the covariate shift problem, in chapter 5 we studied different unsupervised domain adaptation techniques to mitigate its effects. Particularly, we investigated unsupervised adversarial domain adaptation and its limitations versus finetuning (after pre-trainig with synthetic data), and image transformations such as inpainting methods and synthetic image fusion with depth sensor background. While unsupervised adversarial domain adaptation reduces the performance drop on real depth images, we found that simple finetuning on a small annotated dataset of real images surpasses the performance of the rest of the techniques.

In chapter 6 we presented our proposed approach for multi-person 3D pose estimation from depth images. Contrary to state-of-the-art deep learning-based methods that address the task from image crops, our approach decouples the problem in two steps: 1) a 2D pose estimation step leveraging efficient CNN architectures, and 2) a 3D pose regression from lifted 2D coordinates. We introduced a pairwise limb priors to recover 2D undetected body parts due to self-occlusions or detection failures. Our experiments showed that our decoupled approach achieves very competitive performance in both single and multi-person settings and runs efficiently in real-time. Hence, our approach proposes a better efficient alternative for multi-person 3D pose estimation for HRI scenarios than state-of-the-art methods that process heavy voxelized representations with large deep models.

Finally, in chapter 7 we studied human 3D motion prediction in a sequence-to-sequence fashion with a non-autoregressive transformer. state-of-the-art methods usually learn autoregressive RNN models with stacks of LSTM or GRU units, or provide the model with a frequency representation of the sequence. In contrast, our approach prevents the error accumulation that arises in autoregressive decoding by instead predicting elements of the target sequence in parallel from a query pose. We also investigated the use of different architectures to compute embeddings of 3D skeletons and model the spatial dependencies. Given that activities provide context for motion execution, we explored the skeleton-based activity recognition using the transformer encoder self-attention embeddings. Our method shows very promising results achieving similar or better performance in motion prediction of state-of-the-art methods at different temporal horizons.

8.2 Perspectives

Our work opens the door for more research. In this section we discuss some of the limitations of our approaches and propose different directions in that regard.

Modelling 2D Landmark Visibility. Our methods for 2D pose estimation predict incomplete estimates when detection failures and self-occlusions occur. This behaviour arises since our models are trained with ground truth only for visible landmarks. Inferring the location of the invisible or occluded landmarks is a challenging task that normally requires prior knowledge. However, a solution is to predict the locations of all body parts (including occluded) and predict a visibility label for each. To incorporate this idea, our current approach could include the following two extensions: 1) train our CNN models to predict confidence maps for all the body parts (including invisible or self-occluded), 2) incorporate an extra branch to predict binary masks for each body part indicating if that part is visible or not. These extensions require to manually annotate a small set of images with landmark visibility, indicating different semantics for the visibility label, e.g. occluded, visible, out of image bounds.

3D Pose Priors. Our current 3D pose prediction approach presented in chapter 6 use a simple pairwise body part prior to recover missing 2D landmarks. Though we observed good performance by including it, pairwise relationships are a simplified version of modelling the body spatial relationships. A better representation of these relationships should instead consider denser connections between body parts. For example, designing priors that incorporate relationships among entire body extremities comprising different body parts such as shoulders, elbows and wrists.

2D and 3D Landmark Uncertainty Prediction. A concept that we did not explore in our research is the modelling of uncertainties of the body landmark predictions in 2D and 3D coordinates. Since DNN models tend to be overconfident in their estimations, uncertainty information about predictions largely benefits robotics systems to use them in decision making pipelines. In that regard, a thread to follow is to predict a factorized covariance matrix for each landmark and learn body landmark prediction in a maximum likelihood fashion. This approach is more straight forward for learning 3D body landmark in our pose regression setting. However, for the 2D case a more suitable approach is to predict uncertainties from confidence maps and using intermediate CNN representations such as the one presented in (Kumar et al., 2020).

Query Sequence Selection for Motion Prediction. One of the main shortcomings in our motion prediction method is that we feed the transformer decoder with copies of a single query pose. This triggers large error in time steps where the target sequence differ significantly from the query pose. A better strategy is to rely in a process of selection of query poses, for example using the encoder self-attention embeddings. This process could consist on predicting the probability of a pose in the input sequence to be the query pose for position i for the target sequence of length M. Such sequence can be filled by scanning the input sequence copying elements to the most likely position according to the predicted probabilities

Unified Skeleton-RGB Activity Recognition. From our experiments in human activity recognition we observed that a skeleton-based recognition approach performs poorly. We have also seen this behaviour in the different state-of-the-art methods that we analyzed (Shahroudy et al., 2016). Certainly, further context can be extracted from color images that benefits activity recognition such as identifying objects in the image or between person interactions as remarked in (Luvizon et al., 2018). Since this information cannot be extracted from skeletons only, a better solution is to incorporate a second transformer encoder that operates from sequences of image crops. Such transformer encoder predict self-attention embeddings and a class token from which a classifier predicts the activity label. In this setting, we can also explore the fusion of both encoder self-attention embeddings, i.e. the skeleton-based embeddings and the image crops-based embeddings, using linear layers and perform activity classification from the aggregated class tokens. A similar approach has been pursued in multi-modal modelling with transformers like the one presented in (Akbari et al., 2021).

Bibliography

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. (2021). VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In *arXiv:2104.11178v1*.

Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3d human pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455.

Aksan, E., Cao, P., Kaufmann, M., and Hilliges, O. (2021). Spatio-temporal transformer for 3d human motion prediction. *arXiv:2004.08692v2*.

Aksan, E., Kaufmann, M., and Hilliges, O. (2019). Structured prediction helps 3d human motion modelling. In *IEEE International Conference on Computer Vision (ICCV)*.

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). Human pose estimation: New benchmark and state of the art analysis. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures revisited: People detection and articulated pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Ba, S. and Odobez, J. (2006). A study on visual focus of attention recognition from head pose in a meeting room. In *Proc. Workshop on Machine Learning for Multimodal Interaction (MLMI)*, Washington DC.

Bao, Y., Zhou, H., Feng, J., Wang, M., Huang, S., Chen, J., and Li, L. (2020). {PNAT}: Non-autoregressive transformer by position learning. In *arXiv:1911.10677v1*.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Int. Conf. on Machine Learning (ICML)*.

Benzine, A., Chabot, F., Luvison, B., Pham, Q. C., and Achard, C. (2020). Pandanet: Anchor-based single-shot multi-person 3d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6855–6864.

Blender Online Community (2017). Blender - a 3d modelling and rendering package. Build year: 2018.

Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv:1605.07678*.

Cao, Z., Simon, T., Wei, S., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*.

Chen, C., Heili, A., and Odobez, J.-M. (2011a). A joint estimation of head and body orientation cues in surveillance video. In *IEEE CVPR-SISM, Int. Workshop on Socially Intelligent Surveillance and Monitoring, Barcelona*.

Chen, C. and Ramanan, D. (2017). 3d human pose estimation = 2d pose estimation + matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5759–5767.

Chen, C., Yang, Y., Nie, F., and Odobez, J.-M. (2011b). 3d human pose recovery from image by efficient visual feature selection. *Computer Vision and Image Understanding (CVIU)*, 115(3):290–299.

Chen, C., Yu, Y., and Odobez, J.-M. (2015). Head nod detection from a full 3d model. In *Int. Conf. on Computer Vision Workshop, Santiago, Chile,*.

Chen, G., Choi, W., Yu, X., Han, T., and Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems (NIPS)*.

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

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

Ciller, C., De Zanet, S., Kamnitsas, K., Maeder, P., Glocker, B., Munier, F., Rueckert, D., Thiran, J.-P., Bach Cuadra, M., and Sznitman, R. (2017). Multi-channel mri segmentation of eye structures and tumors using patient-specific features. *PLoS ONE*, 12.

CMU Motion Lab (2007). Cmu motion capture database. http://mocap.cs.cmu.edu/. Last checked on April 01, 2021.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, CVPR '05, page 886–893, USA. IEEE Computer Society.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Evangelidis, G., Singh, G., and Horaud, R. (2014). Skeletal quads: Human action recognition using joint quadruples. In *2014 22nd International Conference on Pattern Recognition*, pages 4513–4518.

Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., and Cucchiara, R. (2018). Learning to detect and track visible and occluded body joints in a virtual world. In *European Conf. on Computer Vision (ECCV)*.

Fleet, D. J. (2011). *Motion Models for People Tracking*.

Foster, M. E., Alami, R., Gestranius, O., Lemon, O., Niemelä, M., Odobez, J.-M., and Pandey, A. K. (2016). The mummer project: Engaging human-robot interaction in real-world public spaces. In Agah, A., Cabibihan, J.-J., Howard, A. M., Salichs, M. A., and He, H., editors, *Social Robotics*, pages 753–763, Cham. Springer International Publishing.

Foster, M. E., Craenen, B., Deshmukh, A., Lemon, O., Bastianelli, E., Dondrup, C., Papaioannou, I., Vanzo, A., Odobez, J.-M., Canévet, O., Cao, Y., He, W., Martínez-González, A., Motlicek, P., Siegfried, R., Alami, R., Belhassein, K., Buisan, G., Clodic, A., Mayima, A., Sallami, Y., Sarthou, G., Singamaneni, P.-T., Waldhart, J., Mazel, A., Caniot, M., Niemelä, M., Heikkilä, P., Lammi, H., and Tammela, A. (2019). Mummer: Socially intelligent human-robot interaction in public spaces. In *Proceedings of AI-HRI 2019*.

Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. (2015). Recurrent network models for human dynamics. In *IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 4346–4354, USA. IEEE Computer Society.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. (2018). Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Gui, L.-Y., Wang, Y.-X., Liang, X., and Moura, J. M. F. (2018). Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Guo, H., Wang, G., Chen, X., and Zhang, C. (2017). Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*.

Habibie, I., Xu, W., Mehta, D., Pons-Moll, G., and Theobalt, C. (2019). In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., and Fei-Fei, L. (2016). Towards viewpoint invariant 3d human pose estimation. In *European Conf. on Computer Vision (ECCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*.

He, W., Motlicek, P., and Odobez, J.-M. (2018). Joint Localization and Classification of Multiple Sound Sources Using a Multi-task Neural Network. In *Proc. Interspeech 2018*, pages 312–316, Hyderabad, India.

He, W., Motlicek, P., and Odobez, J. M. (2021). Neural network adaptation and data augmentation for multi-speaker direction-of-arrival estimation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 29:1303–1317.

Hernandez, A., Gall, J., and Moreno, F. (2019). Human motion prediction via spatio-temporal inpainting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7133–7142.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). CyCADA: Cycle-consistent adversarial domain adaptation. In *Int. Conf. on Machine Learning (ICML)*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*.

Hu, P. and Ramanan, D. (2016). Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5600–5609.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv:1602.07360*.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schieke, B. (2016). Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). 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.

Johnson, S. and Everingham, M. (2010). Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11. BMVA Press. doi:10.5244/C.24.12.

Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., and Sheikh, Y. (2017). Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jung, H. Y., Lee, S., Heo, Y. S., and Yun, I. D. (2015). Random tree walk toward instantaneous 3d human pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Kim, H., Lee, S., Lee, D., Choi, S., Ju, J., and Myung, H. (2015). Real-time human pose estimation and gesture recognition from depth images using superpixels and svm classifier. *Sensors*.

Kothari, P., Kreiss, S., and Alahi, A. (2021). Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–15.

Kumar, A., Marks, T. K., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X., and Feng, C. (2020). Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017). Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–44.

Lehrmann, A. M., Gehler, P. V., and Nowozin, S. (2014). Efficient nonlinear markov models for human motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1314–1321.

Li, B., Tian, J., Zhang, Z., Feng, H., and Li, X. (2021). Multitask non-autoregressive model for human motion prediction. *IEEE Transactions on Image Processing*, 30:2562–2574.

Li, C., Zhang, Z., Sun Lee, W., and Hee Lee, G. (2018). Convolutional sequence to sequence model for human dynamics. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, S., Zhang, W., and Chan, A. B. (2015). Maximum-margin structured learning with deep networks for 3d human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conf. on Computer Vision (ECCV)*.

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2016). Unsupervised Domain Adaptation with Residual Transfer Networks. In *Neural Information Processing Systems (NIPS)*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Luvizon, D. C., Picard, D., and Tabia, H. (2018). 2d/3d pose estimation and action recognition using multitask deep learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5137–5146.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451.

Makehuman Online Comunity (2018). Makehuman - open source tool for making 3d characters. Build year: 2018.

Martinez, J., Black, M. J., and Romero, J. (2017a). On human motion prediction using recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Martinez, J., Hossain, R., Romero, J., and Little, J. J. (2017b). A simple yet effective baseline for 3d human pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*.

Martínez-González, A., Villamizar, M., Canévet, O., and Odobez, J. (2018a). Investigating depth domain adaptation for efficient human pose estimation. In *European Conf. of Computer Vision - Workshops, (ECCV)*.

Martínez-González, A., Villamizar, M., Canévet, O., and Odobez, J. (2020a). Efficient convolutional neural networks for depth-based multi-person pose estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):4207–4221.

Martínez-González, A., Villamizar, M., Canévet, O., and Odobez, J. (2020b). Residual pose: A decoupled approach for depth-based 3d human pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Martínez-González, A., Villamizar, M., Canévet, O., and Odobez, J. M. (2018b). Real-time convolutional networks for depth-based human pose estimation. In *Int. Conf. on Intelligent Robots and Systems, IROS*.

Martínez-González, A., Villamizar, M., and Odobez, J. (2021). Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In *IEEE/CVF International Conference on Computer Vision - Workshops (ICCV)*.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. In *Int. Conf. on Learning Representations, (ICLR)*.

Moon, G., Chang, J., and Lee, K. M. (2018). V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European Conf. on Computer Vision*.

Nguyen, L., Odobez, J.-M., and Gatica-Perez, D. (2012). Using self-context for multimodal detection of head nods in face-to-face interactions. In *ACM Int Conf. on Multimodal Interaction (ICMI), Santa Monica*.

Patricia, N., Cariucci, F. M., and Caputo, B. (2017). Deep depth domain adaptation: A case study. In *IEEE Int. Conf. on Computer Vision Workshops, ICCV Workshops*.

Pavlakos, G., Zhou, X., Derpanis, K. G., and Daniilidis, K. (2017). Coarse-to-fine volumetric prediction for single-image 3D human pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P., and Schiele, B. (2016). Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE CVPR*.

Raaj, Y., Idrees, H., Hidalgo, G., and Sheikh, Y. (2019). Efficient online multi-person 2d pose tracking with recurrent spatio-temporal affinity fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Ramakrishna, V., Kanade, T., and Sheikh, Y. (2012). Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision (ECCV)*.

Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, A. J., and Sheikh, Y. (2014). Pose machines: Articulated pose estimation via inference machines. In *ECCV*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Reiter-Palmon, R., Sinha, T., Gevers, J., Odobez, J.-M., and Volpe, G. (2017). Theories and models of teams and group. *Journal of Small Group Research*, 45(5):544–567.

Rogez, G., Weinzaepfel, P., and Schmid, C. (2019a). Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Rogez, G., Weinzaepfel, P., and Schmid, C. (2019b). Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Romero, A., Ballas, N., Ebrahimi Kahou, S., Chassang, A., Gatta, C., and Bengio, Y. (2015). Fitnets: Hints for thin deep nets. In *In Proceedings of Int. Conf. on Representation Learning (ICRL)*.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, Cham. Springer International Publishing.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Shahroudy, A., Liu, J., Ng, T.-T., and Wang, G. (2016). Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sheikhi, S. and Odobez, J. (2015). Combining dynamic head pose and gaze mapping with the robot conversational state for attention recognition in human-robot interactions. *Pattern Recognition Letters*, 66:81–90.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., and Blake, A. (2012). Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Siegfried, R., Yu, Y., and Odobez, J.-M. (2017). Towards the use of social interaction conventions as prior for gaze model adaptation. In *19th ACM International Conference on Multimodal Interaction (ICMI)*, Glasgow.

Sigal, L., Isard, M., Haussecker, H., and Black, M. J. (2011). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision (IJCV)*, 98(1):15–48.

Silberman, N., D., H., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European Conf. on Computer Vision (ECCV)*.

Srinivas, S. and Fleuret, F. (2018). Knowledge transfer with jacobian matching. In *Int. Conf. on Machine Learning (ICML)*.

Sudhakaran, S., Escalera, S., and Lanz, O. (2021). Learning to recognize actions on objects in egocentric video with attention dictionaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Sugiyama, M. and Kawanabe, M. (2012). *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press.

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, X. W., Shang, J., Liang, S., and Wei, Y. (2017). Compositional human pose regression. In *IEEE International Conference on Computer Vision (ICCV)*.

Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A. W. (2012). The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Teixeira, T., Dublon, G., and Savvides, A. (2010). A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, 5:59.

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9.

Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076.

Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2017). Adversarial Discriminative Domain Adaptation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Urtasun, R., Fleet, D., and Fua, P. (2006). 3d people tracking with gaussian process dynamical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CV PR'06)*, volume 1, pages 238–245.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M. J., Laptev, I., and Schmid, C. (2017). Learning from synthetic humans. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Villamizar, M., Martínez-González, A., Canévet, O., and Odobez, J.-M. (2018). Watchnet: Efficient and depth-based network for people detection in video surveillance systems. In *IEEE International Conference on Advanced Video and Signal-based Surveillance*, pages 109–114.

Villamizar, M., Martínez-González, A., Canévet, O., and Odobez, J.-M. (2020). Watchnet++: Efficient and accurate depth-based network for detecting people attacks and intrusion. *Machine Vision and Applications*.

Wang, J., Jiang, W., Ma, L., Liu, W., and Xu, Y. (2018). Bidirectional attentive fusion with context gating for dense video captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, K., Zhai, S., Cheng, H., Liang, X., and Lin, L. (2016). Human pose estimation from depth images via inference embedded multi-task learning. In *ACM on Multimedia Conf.*

Wei, M., Miaomiao, L., and Mathieu, S. (2020). History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision (ECCV)*.

Wei, M., Miaomiao, L., Mathieu, S., and Hongdong, L. (2019). Learning trajectory dependencies for human motion prediction. In *IEEE International Conference on Computer Vision (ICCV)*.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Wu, D., Pigou, L., Kindermans, P.-J., Le, N., Shao, L., Dambre, J., and Odobez, J.-M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Trans. on Patt. Anal. and Machine Intelligence (PAMI)*.

Xu, X., Chen, H., Moreno-Noguer, F., Jeni, L. A., and De la Torre, F. (2021). 3d human pose, shape and texture from low-resolution images and videos. *TPAMI*.

Yang, W., Li, S., Ouyang, W., Li, H., and Wang, X. (2017). Learning feature pyramids for human pose estimation. In *arXiv:1708.01101*.

Yang, Y. and Ramanan, D. (2013). Articulated human detection with flexible mixtures of parts. *IEEE Trans. on Pat. Analysis and Machine Intelligence, (TPAMI)*.

Ye, M., Wang, X., Yang, R., Ren, L., and Pollefeys, M. (2011). Accurate 3d pose estimation from a single depth image. In *IEEE International Conference on Computer Vision*, pages 731–738.

Angel Noé Martínez-González

Contact

Address: Rue Marconi 19
1920 Martigny
Switzerland

Mobile: +41 76 823 3671
E-mail: angelmtzg.09@gmail.com
E-mail: angel.martinez@idiap.ch

Education

11.16-	**École Polytechnique Fédérale de Lausanne (EPFL), Switzerland**. Doctoral Student in Electrical Engineering. Advisor: Dr. Jean-Marc Odobez.
2012-14	**Center for Research in Mathematics (CIMAT), México**. Master in Computer Science. Cumulative Grade Point Average: 92/100.
2007-12	**University of Guanajuato (DICIS), México**. Computer Systems Engineering. Cumulative Grade Point Average: 89/100. Specialization: Artificial Intelligence and Industrial Informatics.

Work Experience

- **08-11.2020** **Google, Zurich, Switzerland**. Research SWE Intern. OCR/Mobile vision team.
- **04-07.2020** **Amazon, Berlin, Germany**. Applied Science Intern. Computer vision/NAC.
- **2012, 2016** **Center for Research in Mathematics (CIMAT), México**. Research and Development Engineer. Machine learning and computer vision techniques for 1) computer graphics and VR gaming, 2) image recognition for Android devices, 3) Robocup 2012.
- **2015** **Intel Corporation (Intel-GDC) México**. Software Engineer. Android OS camera stack.
- **2014** **Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS), Toulouse, France**. Research internship. Intelligent systems for video surveillance.
- **2013-14** **Center for Research in Mathematics (CIMAT), México**. Teaching Assistant. Applied Probability and Statistic MSc course. Instructional responsibilities, grading and teaching R.
- **2012-13** **Center for Research in Mathematics (CIMAT), México**. Science diffusion. Programming sessions of SCRATCH for children at secondary school level.
- **2003-08** **National Institute for Adult Education (INEA), México**. Adult Education. Tutoring sessions and direction of exams for adult certification of basic education.

Research and Publications

Resesearch interests: Human-computer interaction, video analysis, machine learning.

- **Residual Pose: A Decoupled Approach for Depth-Based 3d Human Pose Estimation**. Angel Martinez-Gonzalez, Michael Villamizar, Olivier Canevet and Jean-Marc Odobez. *IEEE/IRSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2020.*
- **WatchNet++: Efficient and Accurate Depth-Based Network for Detecting People Attacks and Intrusion**. Michael Villamizar, Angel Martinez-Gonzalez, Olivier Canevet and Jean-Marc Odobez. *Springer Journal of Machine Vision and Application, (MVAP) 2020.*
- **Efficient Convolutional Neural Networks for Depth-Based Multi-Person Pose Estimation**. Angel Martinez-Gonzalez, Michael Villamizar, Olivier Canevet and Jean-Marc Odobez. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 2019.*
- **WatchNet: Efficient and Depth-based Network for People Detection in Video Surveillance Systems**. Michael Villamizar, Angel Martinez-Gonzalez, Olivier Canevet and Jean-Marc Odobez. *IEEE Int. Conf. on Advanced Video and Signal-Based Processing (AVSS), 2018.*
- **Investigating Domain Adaptation for Efficient Human Pose Estimation**. Angel Martinez-Gonzalez, Michael Villamizar, Olivier Canevet and Jean-Marc Odobez. *European Conference on Computer Vision - Workshops, (ECCV) 2018.*
- **Real-Time Convolutional Neural Networks for Depth-Based Human Pose Estimation**. Angel Martinez-Gonzalez, Michael Villamizar, Olivier Canevet and Jean-Marc Odobez. *IEEE/IRSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2018.*

- **MSc. Thesis: “Motion-Based Camera-Network Topology Inference”**, using probabilistic graphical models, machine learning, and computer vision techniques in a visual surveillance framework. Thesis director: Dr. Jean-Bernard Hayet (CIMAT). Thesis co-director: Dr. Frédéric Lerasle (LAAS-CNRS). Year 2014.
- **Real-time Face Detection Using Neural Networks**. Angel N. Martinez-Gonzalez and Victor Ayala-Ramirez. *IEEE Mexican International Conference on Artificial Intelligence, (MICAI), 2011.*

Skills

- Development C/C++.
- Scripting: Python, Matlab, R.
- Operating systems: Windows (user), Linux (user).

Projects

- **ViZDoom with Reinforcement Learning**. Solving navigation tasks in a 3d FPS game environment for autonomous agents with deep reinforcement learning methods.
- **3d object culling methods for a virtual reality game**: Sophie’s guardian GameCoder Studios, Mexico.
- **Algorithms Analysis and Design Group. Intel Corporation, (Intel-GDC) Mexico**. Informal speeches to cover some topics of algorithm design and analysis for the Intel community.
- **Multi-person visual tracking**. Center for Research in Mathematics (CIMAT), México. Development of a vision system able to track multiple people.
- **3d soccer ball tracking**. Ball tracking for the NAO robot for soccer playing in the Robocup soccer championship.

Awards and Grants

- Scholarship for MSc and PhD studies, National Council for Science and Technology (CONACyT), México, 2012,2016.
- 1st Place by Country University ranking, IEEExtreme programming competition, 2011.
- Academic excellence for the best student of the generation, University of Guanajuato, 2011.
- Outstanding performance student, Merit student award, University of Guanajuato, 2008, 2009, 2010.
- Member of the student committee, Engineering Division, University of Guanajuato, 2009, 2010.

Languages

Spanish: Mother tongue; **English**: Fluent; **French**: Intermediate; **German**: Basic.