UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Ray-based Multi-view 2D to 3D Pose Lifter

Seyed Abolfazl Ghasemzadeh

Thesis submitted in partial fulfillment
of the requirements for the degree of
Ph.D. in Engineering Sciences

Dissertation committee:

Prof. Christophe De Vleeschouwer (UCLouvain, Belgium, advisor)
Prof. Benoît Macq (UCLouvain, Belgium)
Prof. Vincent François-Lavet (VUAmsterdam, The Netherlands)
Prof. Anthony Cioppa (ULiege, Belgium)
Prof. Laurent Francis (UCLouvain, Belgium, chair)

To walk the desert path is better than to sit idly.
And if I do not reach my goal, at least I strove to my capacity.

— Saadi Shirazi

Abstract

Estimating 3D human poses from 2D images is a challenging problem in computer vision, with many applications in different fields such as sports analytics, medical diagnostics, virtual reality, and robotics. Although deep learning methods have advanced 2D pose estimation, estimating 3D poses from monocular 2D images remains challenging, since it is an ill-posed problem due to the ambiguity caused by projective geometry. Multi-view setups offer a potential solution, as they increase the availability of spatial information; however, state-of-the-art methods for multi-view 3D pose estimation are heavily reliant on datasets of multi-view images paired with 3D pose annotations. These datasets are often confined to controlled, laboratory environments, limiting the adaptability of existing models to the diverse and dynamic conditions encountered in real-world scenarios.

This thesis addresses these limitations by presenting a novel, generalizable framework for 3D human pose estimation, which decouples the tasks of 2D pose detection and 3D pose lifting. In the first stage, off-the-shelf 2D pose estimation models are used to independently detect anatomical keypoints in each view. In the second stage, a transformer-based architecture, called Ray-based Multi-view 3D Pose Lifter (RMPL), fuses these 2D keypoints into a unified 3D pose skeleton. Unlike previous methods, RMPL operates entirely in the pose space rather than the image space, and it does not require paired images and 3D pose datasets for training. Instead, the proposed framework leverages synthetic 2D-3D pose pairs, generated using a novel pipeline that simulates the noise characteristics of 2D pose estimators. Moreover, a novel ray-based representation of 2D keypoints is introduced, which enables the model to be invariant to camera calibration parameters. This innovative approach makes it possible to train the RMPL for arbitrary camera setups, ensuring adaptability across diverse deploy-

ment scenarios.

Experimental validation demonstrates the effectiveness and robustness of the proposed framework. Evaluations on widely used benchmarks, including the Human3.6M, CMU Panoptic, and RICH datasets, reveal that the RMPL model achieves significant improvements in 3D pose accuracy, with reductions in Mean Per Joint Position Error (MPJPE) of up to 53% compared to triangulation-based baselines. This reduction is above 60% when comparing to transformer-based solutions using image-based representation of 2D pose. Further, the study includes comprehensive analyses of the impact of visibility, camera configurations, and synthetic data quality on the performance of the proposed method.

By eliminating the dependency on the costly and limited 3D human pose datasets, this research not only advances the state-of-the-art in multi-view 3D human pose estimation but also establishes a scalable and practical framework for deployment in real-world environments. The modular design of the proposed system ensures compatibility with a wide range of 2D pose detectors and camera setups, paving the way for future developments in human motion analysis and its applications.

Gratitude

This thesis is the result of a collaborative effort, and I would like to express my gratitude to all those who contributed to its completion. First and foremost, I would like to thank my advisor *Christophe* for his invaluable guidance, support, and encouragement throughout my PhD journey. He has been a constant source of inspiration and motivation, and I am grateful for the opportunity to work with him. I am also deeply thankful to the members of my thesis committee, Prof. *Benoît Macq*, Prof. *Vincent François-Lavet*, Prof. *Anthony Cioppa*, and Prof. *Laurent Francis*, for their constructive feedback and suggestions that have greatly improved the quality of this work. I am fortunate to have worked in an inspiring research environment at the *ICTEAM* at *UCLouvain*. The journey that led to this thesis would not have been possible without the support of my family and friends. I am especially grateful to my parents, *Hassan* and *Akram*, for their support and encouragement throughout my life even from afar. To my brother *Ali* and my sister *Fatemeh*. To my partner *Alicia* for her love, support, and understanding during the ups and downs of my PhD journey, and for not calling my work "code truc" anymore. To my old friends *Behzad* and *Farhad* who started their PhD journey with me in a different timezone but it felt like we were still together. To my friends from *Posht-e-Bargh* who made my life easier when COVID was trying to make it harder. At *UCLouvain*, thanks to all my colleauges: *Amirafshar*, *Emilie*, *Baptiste*, *Niels*, *Alexander*, *Maxime*, *Victor*, *Vladimir*, *Antoine V.*, *Antoine L.*, *Gabriel*, *Gilles*, *Dani*, *Anne-Sophie*, *Mathieu*, *Florine*, *Benoit*, and others. To *Amirafshar* and *Nafiseh* for helping me settling in Belgium and for their support during my PhD. Thanks to *Hossein*, *Amir M.*, and *Amir P.* for making the last months of my PhD more Persian. I had the pleasure to visit *VITA lab* at *EPFL*, and I would like to thank Prof. *Alexandre Alahi* and his team for their warm welcome and support during

my visit. This thesis represents not only my work but also the collective effort of many individuals who have supported me along the way. I am truly grateful for your contributions and encouragement. Thank you all!

List of Publications

- **S. A. Ghasemzadeh**, A. Alahi, and C. D. Vleeschouwer, "RUMPL: Ray-based Universal Multi-view 2D to 3D Pose Lifter," in *IEEE International Conference on Computer Vision (ICCV)*, 2025. *(Submitted)*

- **S. A. Ghasemzadeh**, A. Alahi, and C. D. Vleeschouwer, "MPL: Lifting 3D Human Pose from Multi-view 2D Poses," in *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV) Workshops*, 2024, https://arxiv.org/abs/2408.10805.

- **S. A. Ghasemzadeh**, G. V. Zandycke, M. Istasse, N. Sayez, A. Moshtaghpour, and C. D. Vleeschouwer, "Deepsportlab: A Unified Framework for Ball Detection, Player Instance Segmentation and Pose Estimation in Team Sports Scenes," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

- E. Lacroix, M. Grandjean, M. Huyberechts, L. Steenbergen, **S. A. Ghasemzadeh**, C. De Vleeschouwer, M. Edwards, "Walking towards the future – Exploring OpenTUG's validity in automatic walking activity analyses and the relationship with cognition in vestibular patients,", in *Behavior Research Methods*, 2025. *(Submitted)*

- A. Maiorca, Adrien Kinart, G. Fletcher, **S. A. Ghasemzadeh**, T. Ravet, C. D. Vleeschouwer, T. Dutoit, "Impact of 3D Cartesian Positions and Occlusion on Self-Avatar Full-Body Animation in Virtual Reality,", in *IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality*, 2025.

- V. Somers, V. Joos, A. Cioppa, *et al.*, "Soccernet Game State Reconstruction: End-to-end Athlete Tracking and Identification on a Min-

imap," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024, pp. 3293-3305.

- A. Maiorca, **S. A. Ghasemzadeh**, T. Ravet, F. Cresson, T. Dutoit, and C. D. Vleeschouwer, "Self-avatar Animation in Virtual Reality: Impact of Motion Signals Artifacts on the Full-body Pose Reconstruction,", arXiv, 2024, https://arxiv.org/abs/2404.18628.

Contents

Contents . . . . . **vii**

List of Figures . . . . . **xi**

1 Introduction **1**
1.1 Motivation 1
1.2 Applications Across Different Domains 3
1.3 Challenges and SOTA 4
1.4 Research Objectives 5
1.5 Thesis Organization 7

2 Background and SOTA **9**
2.1 2D Human Pose Estimation 9
2.1.1 State-of-the-art Methods 9
2.1.2 Multi-task Learning in 2D Human Pose Estimation 11
2.2 3D Human Pose Models 12
2.2.1 Skeleton Models 13
2.2.2 Mesh Models 13
2.3 Single-view 3D Human Pose Estimation 14
2.4 Multi-view 3D Human Pose Estimation 15
2.5 Datasets in 3D Human Pose Estimation 17
2.5.1 Real Datasets 18
2.5.2 Synthetic Data in Computer Vision 20
2.5.3 Synthetic Datasets for 3D Human Pose Estimation 21

3 Proposed Framework for 3D Human Pose Estimation **25**
3.1 Decoupling 2D Pose Estimation and 2D to 3D Pose Lifting . . 25

3.2 Key Properties of the Proposed Framework 27
3.3 3D Ray Representation of Poses 28

4 Ray-based Multi-view 2D to 3D Pose Lifter (RMPL) 31
4.1 Keypoint to Ray Converter 32
4.2 Keypoint-level View Fusion Transformer (VFT) 33
4.3 Pose Fusion Transformer (PFT) 34
4.4 Loss Function 34

5 Synthetic Dataset Generation 35
5.1 Mesh-based Data Generation 35
5.2 Mesh-based Human Pose Dataset Generator (MHP) 36

6 Experimental Validation 39
6.1 Evaluation Metric 39
6.2 Datasets 39
6.2.1 Human3.6M 40
6.2.2 CMU Panoptic 40
6.2.3 RICH 40
6.3 Implementation Details 41
6.4 Comparison to Multi-view Prior Works 41
6.4.1 Human3.6M 42
6.4.2 CMU Panoptic 46
6.4.3 RICH 46
6.5 Comparison to Monocular Baselines 47
6.6 Importance of the 3D Ray Representation 48
6.7 Relevance of Our MHP 49
6.8 Inference speed 51
6.9 Complexity 51
6.10 Execution Time of Data Preparation and Training 51
6.11 Robustness and Occlusion 52
6.11.1 Occlusion 52
6.11.2 Random Keypoint Removal During Training 53
6.11.3 Robustness to Camera Calibration Errors 53
6.12 Qualitative Results 54
6.13 Effect of Max Number of Views 54
6.14 Effect of Camera Geometry on MPJPE 55
6.15 Effect of Changing the 2D Pose Estimator 56

7 Discussion 65

7.1 Insights from the Results 65
7.1.1 Generalizability to In-the-wild Conditions 65
7.1.2 Independence from Camera Calibration 66
7.1.3 Robustness to Occlusions 66
7.1.4 Scalability to Any Number of Views 66
7.1.5 Reduction of 3D Error 66
7.2 Limitations of the Proposed Framework 67
7.2.1 Reliance on Off-the-shelf 2D Pose Estimators 67
7.2.2 Lack of Temporal Information 67

8 Conclusion **69**
8.1 Summary 69
8.2 Perspectives 70

Bibliography **75**

List of Figures

1.1 Examples of 2D and 3D human pose estimation 2
1.2 Comparison between the state-of-the-art and our RMPL 8

2.1 The architecture of PifPaf model 11
2.2 Comparison of top-down and bottom-up approaches for 2D HPE 12
2.3 Comparison of 3D human pose models 13
2.4 Comparison of direct and lifting methods for single-view 3D HPE 15
2.5 Comparison of different approaches for multi-view 3D HPE . . . 17
2.6 Some samples from the Human3.6M dataset 18
2.7 Some samples from the Total Capture dataset 19
2.8 Some samples from the CMU Panoptic dataset 20
2.9 Some samples from the RICH dataset 21
2.10 Some samples from the SURREAL dataset 22
2.11 Examples of AMASS meshes 23

3.1 Comparison of direct and lifting methods for multi-view 3D HPE 26
3.2 3D Ray representation of poses 29

4.1 RMPL's architecture 32

5.1 Mesh-based Human Pose Dataset Generator (MHP) 37

6.1 Examples of inconsistency between Human3.6M and COCO format 44
6.2 Distribution of MPJPE (KP*) error on different datasets 57
6.3 MPJPE comparison when occlusion added 58
6.4 Qualitative results 59
6.5 Qualitative results (cont.) 60
6.6 Qualitative results (cont.) 61

6.7 Visualization of sphere points 62
6.8 Effect of camera composition on MPJPE 63

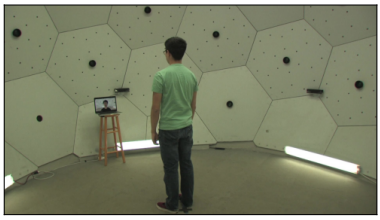
1

Introduction

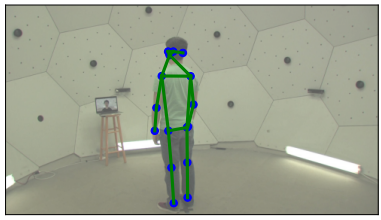
Understanding human motion and posture is a fundamental task in computer vision, with wide range of practical applications in various fields. Human pose estimation (HPE) aims to predict the location and movement of the human joints and limbs (a.k.a human keypoints). While 2D HPE has seen significant advancements with deep learning, 3D HPE remains a challenging problem due to the inherent ambiguity in mapping 2D keypoints to 3D space.

1.1 Motivation

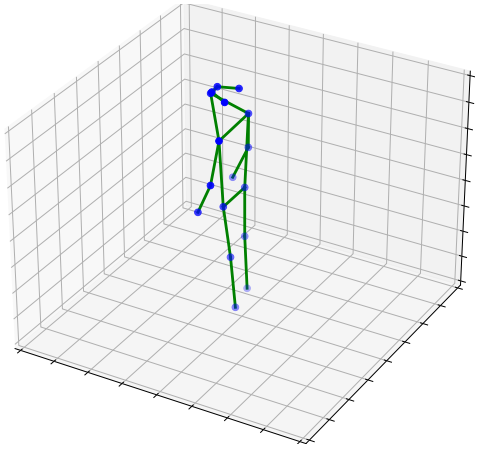
In contrast to 2D HPE, which identifies the spatial configuration of anatomical keypoints on the 2D image plane, 3D Human Pose Estimation aims to infer the three-dimensional locations of these keypoints in space, typically relative to a world or camera-centric coordinate system. While 2D HPE can provide a rough approximation of human posture, it suffers from inherent limitations due to the loss of depth information. For instance, similar 2D projections may correspond to significantly different 3D poses, making the task inherently ambiguous. 3D HPE addresses these ambiguities by explicitly modeling the depth dimension, enabling a more accurate and interpretable representation of the human body in motion. Fig. 1.1 illustrates



(a) 2D image



(b) 2D pose



(c) 3D pose

Fig. 1.1 Examples of 2D and 3D human pose estimation. (a) An image of a person. (b) The 2D keypoints estimated from the image. (c) The estimated 3D keypoints in the metric system. The image and the poses are taken from the CMU Panoptic dataset [1].

this distinction by showing how 3D HPE resolves spatial uncertainty and provides a full-body skeletal reconstruction in 3D space.

The transition from 2D to 3D HPE introduces numerous challenges: it requires dealing with depth ambiguities, occlusions, varying camera viewpoints, scale differences, and the lack of large-scale, high-quality annotated 3D datasets. To mitigate these issues, recent approaches leverage multi-view images, synthetic data generation, parametric human models (*e.g.*, SMPL), and self-supervised learning techniques to improve generalization to in-the-wild scenarios. Despite these challenges, 3D HPE has become an essential technology across diverse domains where accurate motion understanding is critical.

The richer spatial representation provided by 3D HPE opens up new possibilities for analyzing, interpreting, and interacting with human motion in a wide range of settings. By capturing how humans move and orient themselves in real-world environments, 3D HPE supports tasks that require high spatial fidelity, physical plausibility, and temporal consistency. It allows for:

- Depth-aware motion capture, enabling systems to track human motion in cluttered scenes with complex backgrounds.
- Accurate interaction modeling, essential for human-computer interaction, VR/AR applications, and physical simulation.
- Personalized behavioral understanding, such as identifying movement patterns specific to individuals which can be useful for person re-identification.

1.2 Applications Across Different Domains

Sports and Biomechanics: 3D HPE enables detailed biomechanical analysis of athletes by reconstructing body kinematics over time. This facilitates performance optimization, technique refinement, and early detection of injury-prone movements. In team sports, such as soccer or basketball, it also supports strategy analysis by modeling players' spatial relations and dynamic formations.

Surveillance and Security: In video surveillance, 3D HPE enhances human tracking, gait analysis, and re-identification by utilizing depth-aware motion cues. This makes it more robust to camera placement changes, crowded scenes, and occlusions. Systems leveraging 3D pose data can better predict motion trajectories and detect unusual behaviors or emergency events, such as sudden collapses or erratic movements.

Healthcare and Rehabilitation: 3D pose estimation plays a key role in medical monitoring and assisted living. It enables non-intrusive, continuous tracking of patient movements, useful for detecting falls among the elderly, monitoring motor impairments, or quantifying recovery progress in rehabilitation scenarios. By analyzing joint trajectories and symmetry, clinicians can gain insights into neurological disorders, post-surgical mobility, or physical therapy outcomes.

Human-Robot Interaction (HRI): In robotics and related fields, 3D HPE helps autonomous agents understand human intentions and respond appropriately in shared spaces. Robots equipped with 3D perception can adapt their actions based on human pose dynamics, making interactions safer and more intuitive in domestic, industrial, or service environments.

Virtual and Augmented Reality: Accurate 3D human pose tracking is fundamental for immersive experiences in VR/AR. It allows for real-time avatar control, gesture recognition, and motion-based interaction in virtual environments. 3D HPE also supports realistic animation generation and performance capture for entertainment and game development.

Behavioral and Social Analysis: Beyond physical motion, 3D HPE facilitates high-level behavior understanding. In social sciences or marketing research, it can help analyze group interactions, body language, or emotional expression through posture and gesture cues. In neuroscience, it assists in quantifying subtle motor anomalies for early diagnosis of degenerative conditions.

1.3 Challenges and SOTA

Despite significant advancements in 2D HPE achieved based on deep neural networks [2], extending these methods to accurately estimate 3D human poses presents several challenges. The task of predicting 3D keypoints from 2D images remains inherently difficult due to the ill-posed nature of the problem, where multiple 3D poses can correspond to the same 2D projection. This ambiguity, caused by projective geometry, complicates the estimation of the 3D poses particularly in monocular setups.

One of the major challenges in monocular 3D HPE is handling occlusions and oblique viewpoints. Occlusions occur when parts of the human body are not visible in the image, either due to self-occlusions or external occlusions by objects or other people. Oblique viewpoints refer to the camera being positioned at an angle relative to the human subject, leading to perspective distortions that make it difficult to accurately estimate the 3D pose. These challenges limit the reliability of monocular systems, making multi-view setups more appealing for 3D HPE [3].

Multi-view camera setups have shown promises in improving 3D HPE by providing additional information from different viewpoints, which can help resolve ambiguities and improve accuracy [4–10]. However, current

multi-view methods often struggle to generalize to real-world scenarios. Many existing approaches rely heavily on training data collected in controlled laboratory environments (a.k.a 'in-the-lab'), with limited variability in scene appearance and camera placement [1, 6]. These datasets typically contain multi-view images paired with 3D pose annotations, which are expensive and time-consuming to collect, and fail to capture the diversity of real-world scenarios.

Moreover, most existing methods implement feature fusion in the image feature space [3, 9], limiting their robustness when the viewpoints or camera configurations change. These methods are often rigid and require retraining or fine-tuning for new camera setups, making them impractical for real-world deployment.

In summary, current state-of-the-art methods face challenges related to:

- Ill-posed nature of the problem in monocular setups.
- Handling occlusions and oblique viewpoints.
- Generalizing to real-world scenarios.
- Dependency on multi-view images paired with 3D poses.

These challenges highlight the need for more robust, flexible, and scalable solutions for 3D HPE.

1.4 Research Objectives

The primary objective of this research is to develop a universal 3D human pose estimation framework that addresses the key limitations of the existing methods, focusing on generalizability, robustness, and scalability to ensure practical deployment in real-world scenarios. The proposed solution is designed to achieve the following goals:

1. **Generalizability to in-the-wild conditions:**
   Develop a 3D pose estimation system capable of handling never-seen scenes, without being constrained by controlled lab environments.
2. **Capability to adapt to arbitrary camera topologies:**
   Design a method that can adapt to any camera setup without requiring retraining or fine-tuning, ensuring flexibility in real-world deployment scenarios.

3. **Robustness to occlusions:**
   Ensure the system can handle partial visibility of human keypoints due to occlusions or oblique viewpoints by implementing a transformer-based architecture that can effectively fuse information from multiple views, even when some keypoints are missing.

4. **Scalability to any number of views:**
   Create a solution that can adapt to any number of input views during inference without requiring retraining or fine-tuning. The system should be able to handle both sparse and dense multi-view setups, ensuring versatility in various applications.

5. **Reduction of 3D error:**
   Aim to minimize the 3D pose estimation error by improving the accuracy of the 3D pose lifter, particularly in scenarios with challenging camera configurations or varying distances from the subject.

In pursuit of these objectives, the proposed framework decouples 3D HPE from image data by:

- Utilizing off-the-shelf 2D pose detectors for robust keypoint detection.

- Implementing a novel architecture that processes 3D ray features instead of image-based features, ensuring independence from specific camera setups and improving generalizability.

- Training the proposed architecture using synthetic 2D-3D pose pairs, reducing the dependency on multi-view images paired with 3D poses.

Fig. 1.2 illustrates the proposed framework, called Ray-based Multi-view 2D to 3D Human Pose Lifter (RMPL), highlighting the decoupling of 2D pose estimation and 3D pose lifting, as well as the fusion of information from multiple views to predict a 3D pose. This approach enables the training of RMPL for arbitrary acquisition setups thanks to its 3D ray representation and synthetic 2D-3D pose pairs, enhancing its generalizability and robustness in real-world scenarios. By achieving these objectives, this research delivers a scalable, flexible, and robust 3D human pose estimation system suitable for in-the-wild deployment, significantly improving upon current state-of-the-art methods.

1.5 Thesis Organization

This manuscript is organized as follows. Chapter 2 presents the background and some state-of-the-art methods related to 3D human pose estimation. Chapter 3 describes the proposed framework for 3D human pose estimation and the key insights that led to its design. Chapter 4 presents the Ray-based Multi-view 2D to 3D Pose Lifter (RMPL) and its components. Chapter 5 explains how the synthetic dataset of 2D-3D pose pairs is generated for training the 2D to 3D pose lifter. Chapter 6 presents the experimental validation of the proposed framework and the results obtained. Chapter 7 discusses the results and the limitations of the proposed framework. Finally, Chapter 8 concludes the manuscript and discusses future work.

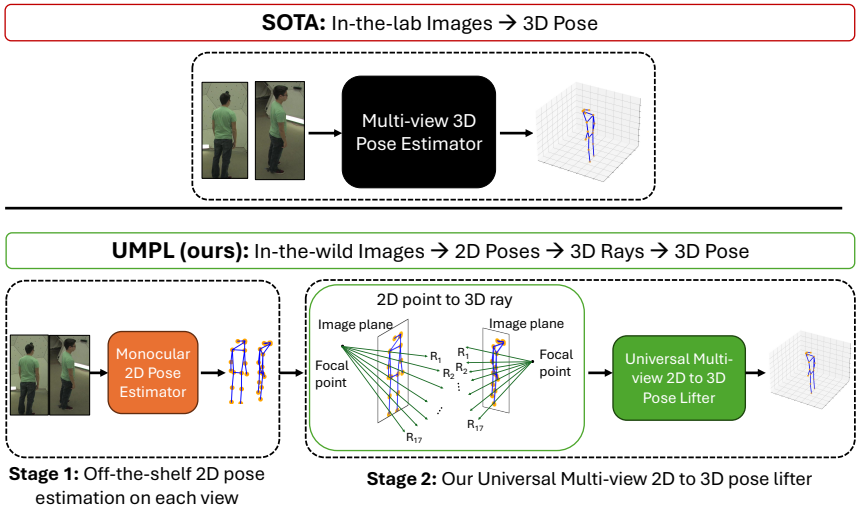


Fig. 1.2 Comparison between the state-of-the-art and our RMPL. On top, the approach used by most prior works. It takes images as input and directly predicts a 3D pose. In the bottom, our approach splits the process into two stages: 1) Performing 2D pose estimation on each view, and 2) Fusing the information from all views to predict a 3D pose. As a key advantage, our RMPL can be trained for arbitrary acquisition setups, using synthetic 2D-3D pairs of skeletons. This is thanks to the 3D ray representation of 2D keypoints, which omits the dependency of the network on the camera calibration, making it more flexible and robust in real-world scenarios. In contrast, prior methods build on images-3D pose correspondences, which are only available for specific scenes and acquisition conditions, thereby penalizing the generalization capabilities of the trained models.

2

Background and SOTA

This chapter will introduce the general concepts and methods for a deeper understanding of the remaining of this manuscript. It first introduces the state-of-the-art methods on 2D HPE and multi-task learning in pose estimation. Then, it presents 3D human pose models used to represent a human in 3D space. It discusses the single-view 3D HPE methods followed by the multi-view 3D HPE methods. Finally, it introduces the conventional datasets used to train and test 3D HPE.

2.1 2D Human Pose Estimation

2D HPE is the task of localizing the body parts (*e.g.*, head, hips, feet, and hands) and joints (*e.g.*, knees and elbows) of human beings in the images. Through this manuscript, we refer to these body parts and joints as keypoints.

2.1.1 State-of-the-art Methods

All recent HPE methods are based on Neural Networks such as Convolutional Neural Networks (CNN) and Transformers [9, 11–15]. Regarding data, there exists large 2D HPE datasets, such as COCO dataset [16], available for training 2D HPE. This makes the 2D HPE methods highly robust

to new scenes and diverse environments, further enhancing their adaptability and generalization capabilities.

2D HPE approaches can be divided into two groups: *top-down* [17–22] and *bottom-up* [15, 23–29]. Top-down methods first perform a human detector and then locate the body parts within every detected bounding-box. Sun *et al.* [30] introduced a High-Resolution Network (HRNet) that maintains high-resolution representations through the whole CNN network. Xu *et al.* [31] proposed a Vision Transformer (ViT) based model for 2D HPE. The model is trained on the COCO dataset [16] and achieves state-of-the-art performance.

Bottom-up methods, on the contrary, first estimate body parts followed by the association of keypoints in skeletons. Associating the detected keypoints into individual instances, however, requires specific additional information. Bottom-up approaches work on a probabilistic map called heatmap that estimates the likelihood of each pixel corresponding to a particular keypoint. The exact location of the keypoint is then estimated by finding the local maximum value in the heatmap. Although the bottom-up approaches are less complex and faster, there is a substantial drop in accuracy compared to top-down approaches [15].

OpenPose [26, 32] model revolutionized the bottom-up approaches by showing that a non-parametric representation called Part Affinity Field (PAF) can be learned to associate body parts with individual human instances. Newell *et al.* [25] used the notion of associative embedding, which can be thought as a vector representing each keypoint's cluster. In particular, keypoints with similar embedding vectors are assigned to the same skeleton. PersonLab [24] integrates the HPE with instance segmentation. It learns to predict relative displacement of body parts, *i.e.*, reminiscent of the idea of spatial embedding [33], allowing to group body parts into human skeleton. By combining and extending the use of short-range offset vectors (as in PersonLab) and PAF (as in OpenPose), Kreiss *et al.* [15] presented PifPaf framework, which reaches excellent performance for crowded scenes. PifPaf introduces two key innovations: the Part Intensity Field (PIF) and the Part Association Field (PAF)—not to be confused with the OpenPose PAFs. The PIF provides high-precision keypoint localization by regressing the exact position and scale of joints, while the new PAF variant predicts associations between keypoints via mid-range offset vectors, optimized for difficult occlusion scenarios. Unlike previous approaches, PifPaf decouples localization from association, leading to a more robust formulation for multi-person pose estimation in challenging environments such as street

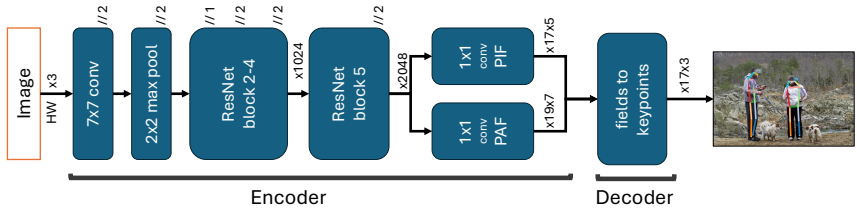


Fig. 2.1 The architecture of PifPaf [15] model. The input is an image of size (H, W) with three color channels, indicated by "$\times 3$". The encoder produces PIF and PAF fields with 17×5 and 19×7 channels, respectively. An operation with stride two is indicated by "//2". The decoder is a program that takes the PIF and PAF fields and produces the final pose. Each joint is represented by an x and y coordinate and a confidence score.

scenes. Fig. 2.1 shows the architecture of PifPaf model as an example of a state-of-the-art 2D human pose estimator. It takes as input an image of size (H, W) with three color channels. The encoder, composed of neural network blocks, produces PIF and PAF fields with 17×5 and 19×7 channels, respectively. The number 17 refers to the 17 keypoints defined by the COCO [16] dataset. Finally, the decoder, a program that takes the PIF and PAF fields, produces the final pose.

Fig. 2.2 illustrates an example of the top-down and bottom-up approaches for 2D HPE. The top-down approach first detects the human instances (bounding boxes) and then estimates the keypoints within the bounding boxes. The bottom-up approach first estimates the keypoints (PIFs) and then groups them into human instances using PAFs.

2.1.2 Multi-task Learning in 2D Human Pose Estimation

There have been several attempts to combine the tasks of HPE and other computer vision tasks such as action recognition [35, 36]. This is motivated by the fact that the tasks are related and can benefit from each other. Moreover, multi-task learning can help lower the computational cost of the model. For instance, the task of HPE can be improved by leveraging the information from the task of human instance segmentation while decreasing the computational cost. This is the motivation behind our work in DeepSportLab [37] with the goal to train a single network to perform 2D human pose estimation, human instance segmentation, and ball de-

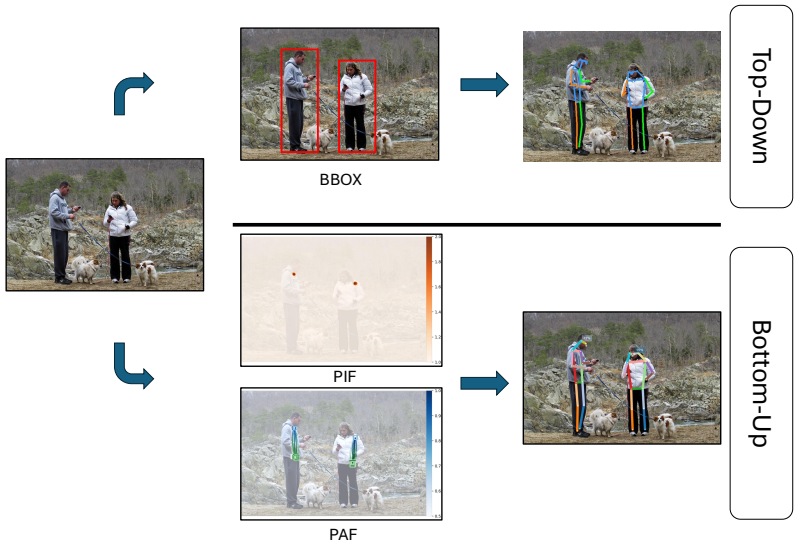
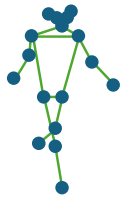


Fig. 2.2 Comparison of top-down and bottom-up approaches for 2D HPE. Top-down methods first detect the human instances (bounding boxes) and then estimate the keypoints within the bounding boxes. The sample skeleton shown for top-down method is depicted using HRNet [30] in MMPOSE [34] framework. Bottom-up methods first estimate the keypoints (PIFs) and then group them into human instances using PAFs. The example of PIF and PAF and the bottom-up skeleton are depicted using PifPaf [15].

tection in team sports scenes. DeepSportLab adopts the PIF part of the PifPaf [15] framework for predicting the location of the keypoints. Regarding the association of the body part keypoints to the poses, it follows the Panoptic-DeepLab's [38] formalism. DeepSportLab leverages the predicted player instance masks to group the keypoints into player poses. Please refer to [37] for more details.

2.2 3D Human Pose Models

To model the human body in 3D, several models have been proposed. These models are used in 3D HPE tasks to represent the human body in



(a) COCO skeleton model

(b) SMPL mesh model

Fig. 2.3 Comparison of 3D human pose models. Skeleton models represent the human body as a set of joints connected by bones. Mesh models represent the human body as 3D vertices.

3D space. The 3D human pose models can be categorized into two main categories: skeleton models and mesh models.

2.2.1 Skeleton Models

In these models, each node represents a joint and the edges roughly represent the bones. One example of this kind of approach is the Pictorial Structure Model (PSM) by Fischler and Elschlager [39]. In this manuscript, we consider the skeleton model compatible with COCO dataset [16] format shown in Fig. 2.3a. This model consists of 17 keypoints representing the human body. The keypoints are the head, neck, shoulders, elbows, wrists, hips, knees, and ankles. This model is widely used in 3D HPE tasks.

2.2.2 Mesh Models

Mesh models consist of complete reconstruction of the human shape. These models are more detailed and can be used for more complex tasks due to the availability of the shape information. One of the most popular mesh models is the SMPL [40] model. The mesh is learned from a large dataset of 3D scans of human bodies and can be adapted to a set of pose and shape parameters. The SMPL-X [41] model extends the SMPL model by adding more shape parameters and a more detailed face model. There exist some works trying to either fit the SMPL model to the 2D keypoints using conventional optimization methods [41, 42] or predict the SMPL parameters directly from the image [43–45]. However, the optimization-based methods are very slow and require a lot of computational resources [43]. The

learning-based methods, on the other hand, while being faster, are limited to the in-the-lab scenes [44, 45]. In this manuscript, we only use mesh models for 3D rendering and visualization purposes. Fig. 2.3b shows a 3D rendered SMPL human body using the open-source software **Body Visualizer** [41] and **Trimesh** [46].

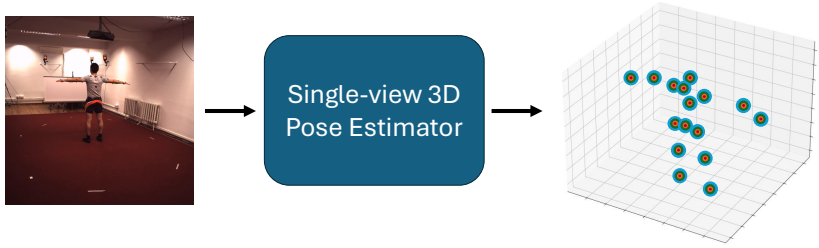
2.3 Single-view 3D Human Pose Estimation

3D HPE is the task of estimating the 3D coordinates of the human keypoints from a single or multiple images. This section focuses on the single-view 3D HPE methods. As explained in Chapter 1, this problem is ill-posed, by nature, as there are multiple 3D poses that can project to the same 2D pose. This makes it very challenging to estimate the 3D pose from a single image. However, there have been several attempts to solve this problem using deep learning methods.

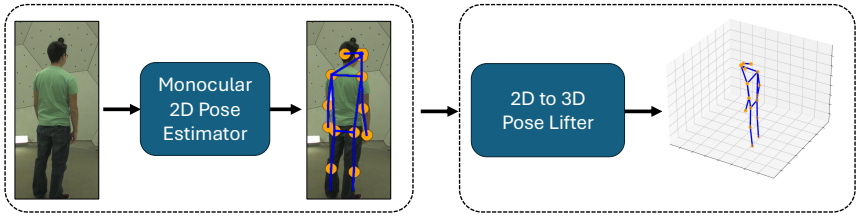
The current state-of-the-art in single-view 3D HPE can be categorized into two main categories. The first category consists of methods that infer the 3D coordinates directly from images using CNNs. Fig. 2.4a depicts the overall pipeline of these methods. They are complex and, more importantly, limited to the in-the-lab scenes [47, 48]. This is due to the fact that they are trained on motion capture datasets, which are scarce and do not cover the diversity of real-world scenes. This problem of the availability of the datasets, and induced lack of generalization to in-the-wild scenes, is covered in more details in Section 2.5.

The second category includes methods that use high quality 2D pose estimators and turn the 2D poses into 3D poses with deep neural networks, *e.g.*, convolutional, recurrent, or transformers. Fig. 2.4b depicts the overall scheme of this category of methods. Martinez *et al.* [49] introduced a simple yet effective method for lifting 2D poses to 3D. They used a fully connected network to predict the 3D pose from the 2D pose. Fang *et al.* [50] further improved the lifting process by introducing a grammar model that enforces the kinematic constraints of the human body.

This category also includes methods that rely on temporal consistency to lift the 2D poses to 3D. Pavllo *et al.* [51] proposed a method that uses a temporal convolutional network to predict the 3D pose from the 2D poses through a sequence of frames. PoseFormer [52] introduced a transformer-based method that uses the temporal information to predict the 3D pose



(a) Direct single-view 3D HPE



(b) Lifting 2D to 3D pose

Fig. 2.4 Comparison of direct and lifting methods for single-view 3D HPE. Direct methods estimate the 3D pose directly from the image. Lifting methods first estimate the 2D pose and then lift it to 3D using either a conventional optimization method or a learning-based approach.

from the 2D poses. Later, Zhang *et al.* [53] pointed out that the paradigm of PoseFormer might neglect distinct temporal patterns for each joint due to being spatial-then-temporal. Instead, they adopt spatial-temporal layers for fine-grained joint-wise feature extraction.

They are simple, fast, and can be trained on separate datasets for the 2D HPE and the lifting process making them more accurate in real-world scenes compared to the works from the first category [35,53–55]. However, due to working with a single view, they are unable to predict the 3D poses in world coordinates and are limited to the camera coordinates.

2.4 Multi-view 3D Human Pose Estimation

To overcome the limitations of single-view 3D HPE, multi-view 3D HPE methods have been proposed. These methods use multiple views of the

same scene to estimate the 3D pose of the human body. The main idea is to fuse the information from all views to estimate the 3D pose more accurately and overcome the ambiguities of single-view 3D HPE.

The most common approach is to estimate the 2D pose in each view and then geometrically triangulate the 3D pose from the 2D poses. In this manuscript, we refer to this approach as the *triangulation* approach (see Fig. 2.5). However, this approach is limited by the accuracy of the 2D pose estimation and the calibration of the cameras. In triangulation, for the 3D estimation of each keypoint, we need confident 2D detections in at least two views to triangulate the 3D position. Such accurate 2D detections are not always available in practice, especially in crowded scenes or when the person is partially occluded.

In recent years, there has been an increasing interest in dealing with multi-view 3D HPE by fusing the image features from all views [3, 8, 9, 56, 57]. This approach is depicted in Fig. 2.5 as the *image feature fusion* approach. Epipolar Transformers [56] fuses the features of one pixel in one view with features along the corresponding epipolar line of the other views. Cross View Fusion [57] learns a fixed attention matrix for fusing features in all other views. TransFusion [8] applies transformers globally to fuse features of the reference views. Learnable-triangulation [3] aggregates the features from different views in a volume by unprojecting the features using the camera calibration parameters. Finally, PPT [9] utilizes a transformer-based approach for 2D pose estimation and makes use of the attention matrix weights to prune the unimportant visual tokens. However, all these methods require multi-view images paired with annotated 3D pose, making them suited to images corresponding to the scene and acquisition setup used at training but vulnerable to novel deployment conditions.

Another approach is to estimate the 3D pose by fusing the features of the 2D keypoints from all views. This approach is depicted in Fig. 2.5 as the *2D-keypoint feature fusion* approach [58–61]. The closest work to this manuscript in this category is probably [61] which proposes concatenating all the joints' 2D coordinates from all views and treating them as a single input to a fully connected network trained to predict the 3D pose in world coordinates. However, we show in Section 6.4 that a fully connected network is prone to strong over-fitting and can fall short in accuracy when facing unseen data. In this manuscript, in Chapter 4, we propose a transformer-based approach for multi-view 3D HPE that fuses the 2D keypoints from all views to estimate the 3D pose in world coordinates.

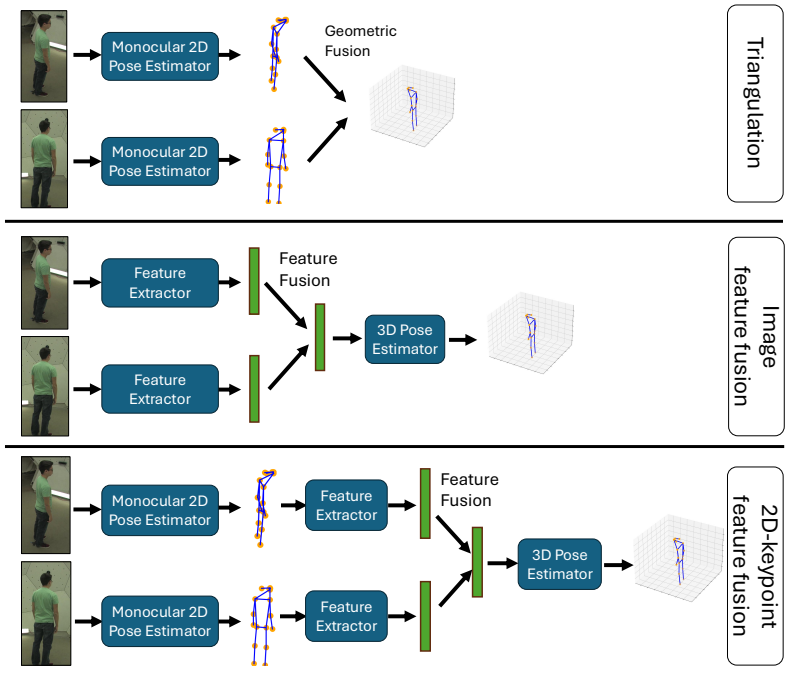


Fig. 2.5 Comparison of different approaches for multi-view 3D HPE. The triangulation approach estimates the 2D pose in each view and then triangulates the 3D pose using the camera calibration parameters. The image-based fusion approach estimates the 3D pose directly from the multi-view images by fusing the image features. The 2D-keypoint-based fusion approach estimates the 3D pose by fusing the 2D keypoints from all views.

We show in Chapter 6 that our approach is more robust and generalizable compared to the existing methods.

2.5 Datasets in 3D Human Pose Estimation

This section introduces some of the most widely used datasets in 3D HPE. First, it introduces some datasets that are created by capturing real images in controlled environments. Then, it talks about the importance of using synthetic data in computer vision. Finally, it introduces some datasets that

Fig. 2.6 Some samples from the Human3.6M dataset [6].

are created by rendering synthetic images.

2.5.1 Real Datasets

3DPW

The 3D Poses in the Wild [62] dataset is the first dataset in the wild with accurate 3D poses for evaluation. 3DPW is the first one that includes video footage taken from a moving phone camera. This dataset includes 60 video sequences of 18 different 3D models in different clothing. It contains 2D pose annotations and 3D poses obtained with their method which leverages video and IMU data. While the dataset is diverse and challenging, it only provides a single view of the scene.

Human3.6M

Human3.6M [6] is the most widely used public dataset for single-person 3D HPE. It consists of 11 professional actors performing 17 actions such as talking, walking, and sitting. The videos are recorded from 4 different views in an indoor environment. In total, this dataset contains 3.6 million video frames with 3D ground truth annotations, collected by an accurate marker-based motion capture system. Despite its popularity, the dataset has several limitations such as the limited number of subjects, the limited number of actions, and the controlled environment. Fig. 2.6 shows some samples from the Human3.6M dataset.

HumanEva

This dataset contains 7 calibrated video sequences (4 grayscale and 3 color) that are synchronized with 3D body pose data [63]. The dataset includes 4 subjects performing 6 actions such as walking, jogging, and gesturing.

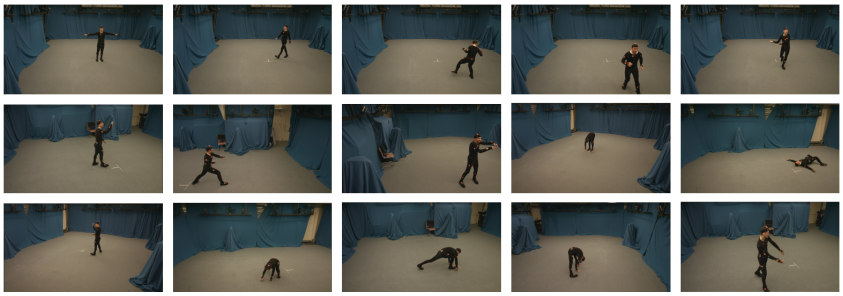


Fig. 2.7 Some samples from the Total Capture dataset [64].

The dataset contains training, validation, and testing sets. However, the dataset is limited to in-the-lab scenes and does not cover the diversity of real-world scenes.

Total Capture

Total Capture [64] contains a number of subjects performing varied actions and viewpoints. It was captured indoors in a volume measuring roughly $8 \times 4\ m$ with 8 calibrated HD video cameras at $60Hz$. There are 4 male and 1 female subjects each performing four diverse performances, repeated 3 times: ROM, Walking, Acting and Freestyle. There is a total of almost 1.9M frames of synchronized video, IMU and Vicon data. Total Capture also suffers from the limitation of being in-the-lab and not covering the diversity of real-world scenes. Fig. 2.7 shows some samples from the Total Capture dataset.

CMU Panoptic

CMU Panoptic is a more recent multi-view public dataset made available by the Carnegie Mellon University. The dataset contains Full-HD video streams of 40 subjects captured/observed from up to 31 cameras located in a dome. 3D ground-truth pose annotations are generated via triangulation using all camera views. While offering a large number of subjects and views, the dataset is still limited to in-the-lab scenes and does not cover the diversity of real-world scenes. Fig. 2.8 shows some samples from the CMU Panoptic dataset.

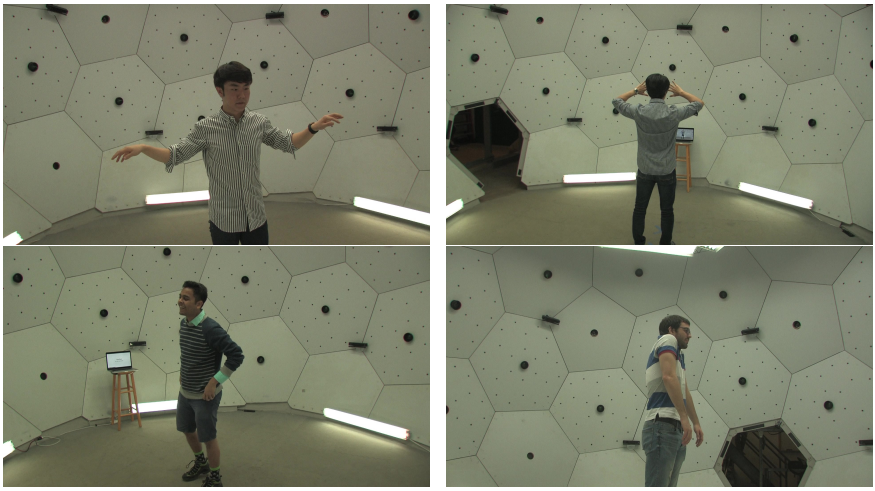


Fig. 2.8 Some samples from the CMU Panoptic dataset [1].

RICH

Real scenes, Interaction, Contact, and Humans (RICH) [65] is a dataset of multi-view videos with accurate bodies and 3D scans. It uses marker-less motion capture systems to annotate the 3D poses. This dataset is captured outdoor using between 5 to 8 cameras per scene covering different actions. It provides 3D meshes in form of SMPL-X [41]. Fig. 2.9 shows some samples from the RICH dataset.

2.5.2 Synthetic Data in Computer Vision

Synthetic data has become a useful resource in computer vision in recent years offering a way to generate large amounts of labeled data for training. By generating synthetic data, one can control the environment, the camera parameters, the lighting, and the ground truth annotations. This allows them to create datasets that are more diverse and more challenging while it might be expensive or even impossible to collect such data in real-world scenarios. By generating artificial data through methods like 3D graphics modeling and generative adversarial networks (GANs) [66], researchers can create diverse and annotated images that enhance model performance and generalization.

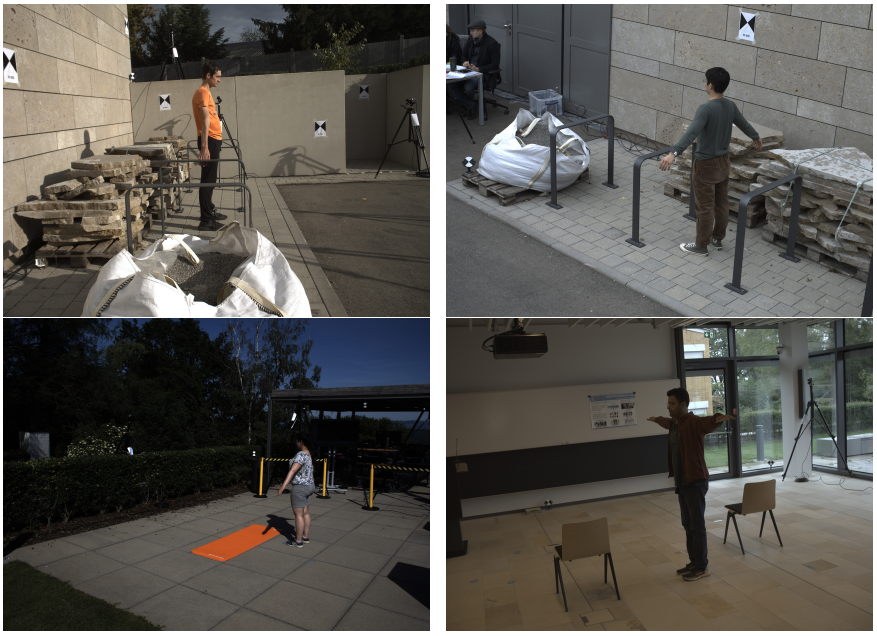


Fig. 2.9 Some samples from the RICH dataset [65].

2.5.3 Synthetic Datasets for 3D Human Pose Estimation

SURREAL

Synthetic hUmans foR REAL tasks (SURREAL) [67] is a large-scale single-view 3D dataset containing 6M frames of synthetic humans. The dataset pairs RGB videos with depth, body parts, optical flow, 2D/3D pose, and shape ground truths. The synthetic bodies are created using the SMPL [40] body model, whose parameters are fit by the MoSh [68] method given raw 3D MoCap marker data. Fig. 2.10 shows some samples from the SURREAL dataset.

AMASS

AMASS [68] is an archive of motion capture datasets in the format of SMPL parameters [40]. This dataset provides a variety of human shapes collected from many datasets and represented with standard 3D mesh models, convertible to SMPL vertices. AMASS contains 3D human motion capture data from more than 15 different datasets, including Human3.6M [6], CMU Panoptic [1], and Total Capture [64]. It provides a large number of 3D poses

Fig. 2.10 Some samples from the SURREAL dataset [67].

with accurate annotations in SMPL format. This dataset is not actually synthetic, but it is used to generate synthetic data for training. Fig. 2.11 depicts some 3D rendered samples from the AMASS dataset.

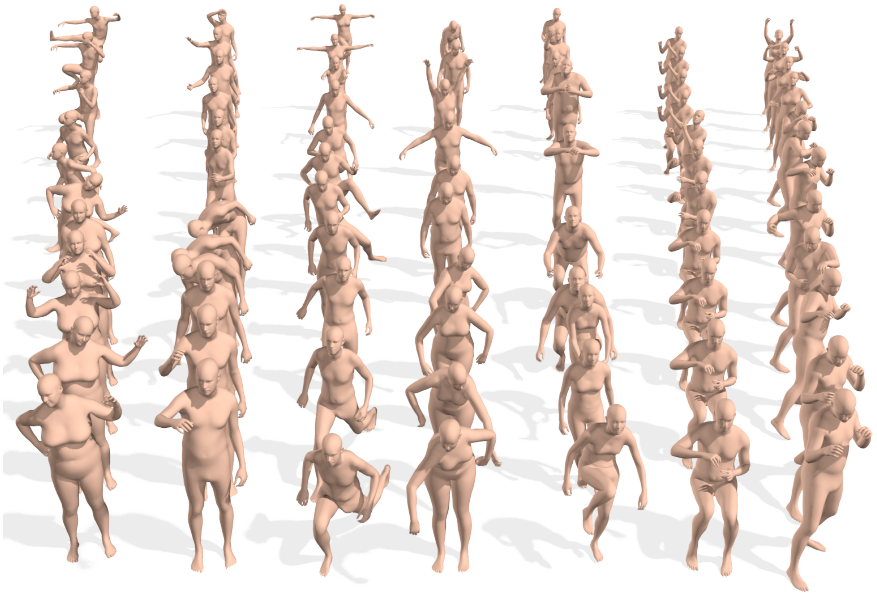


Fig. 2.11 AMASS is a large database of human motion capture data in SMPL format [68].

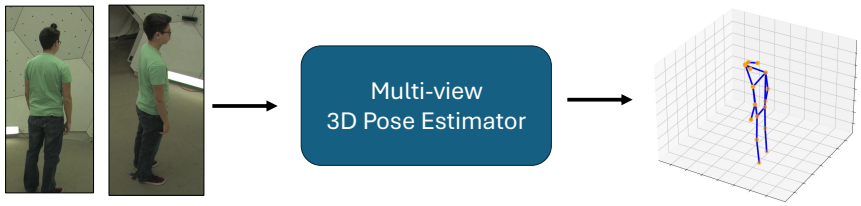
3

Proposed Framework for 3D Human Pose Estimation

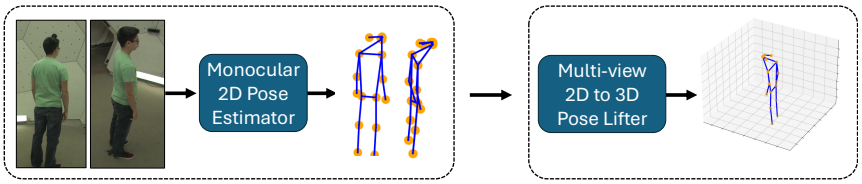
This chapter presents the proposed framework for 3D human pose estimation. The framework is designed to address the limitations of the existing methods, which are discussed in Chapter 2. The proposed framework is based on the observation that the 3D human pose estimation problem can be decomposed into two sub-problems: 2D pose estimation and 2D to 3D pose lifting. This chapter describes, in more details, the key insights that led to the design of the proposed framework.

3.1 Decoupling 2D Pose Estimation and 2D to 3D Pose Lifting

As discussed in Section 2.4, multi-view 3D HPE methods work by fusing information from multiple views. This fusion process is typically done in image-space, where the 2D image features from each view are combined to produce a 3D pose estimate (a.k.a. direct methods). However, this approach has several limitations, including the need for multi-view images paired with 3D pose annotations. As explained in Section 2.5, such datasets are expensive to collect and are limited to in-the-lab conditions making the framework not deployable to in-the-wild images.



(a) Direct multi-view 3D HPE



(b) Lifting 2D to 3D pose

Fig. 3.1 Comparison of direct and lifting methods for multi-view 3D HPE. Direct methods estimate the 3D pose directly from multi-view by image features fusion using transformers or other methods. Lifting methods first estimate the 2D pose in each view and then lift it to 3D using geometric fusion.

To address these limitations, the proposed framework decouples the 3D human pose estimation problem into two sub-problems: 2D pose estimation and 2D to 3D pose lifting. This can be translated to fusing the information from different views in the *2D-keypoint feature space* (see Section 2.4). Fig. 3.1 illustrates the difference between direct and lifting methods for multi-view 3D HPE.

The most important advantages of decoupling 2D pose estimation and 2D to 3D pose lifting are that (i) robust 'in-the-wild' 2D pose detectors exist, and (ii) only pairs of 2D and 3D poses are needed to train our 2D to 3D pose lifter, eliminating the need for images paired with 3D poses. Creating synthetic 2D-3D pose pairs is much cheaper than collecting multi-view images paired with 3D poses, making the proposed framework more scalable and deployable to in-the-wild images. Chapter 5 describes how realistic pairs of 2D-3D poses can be generated for arbitrary viewpoints using the AMASS human shape dataset [68]. The noise associated with off-the-shelf

2D pose estimation models is taken into account by deriving the 2D poses from images rendered from the AMASS dataset (explained in Section 5.2).

3.2 Key Properties of the Proposed Framework

To design a robust and deployable 3D human pose estimation framework, three key properties are considered: (i) the ability to recover occluded keypoints, (ii) the ability to generalize to new scenes, and (iii) the compatibility with new camera setups. Table 3.1 compares the proposed framework with existing methods based on these properties.

The ability to recover occluded keypoints is crucial for robust 3D human pose estimation, as occlusions are common in real-world scenarios. This property is the key issue of triangulation-based methods, which are unable to recover occluded keypoints. This is because triangulation requires the visibility of the keypoints in all input views (or at least in two). Learning-based methods, however, can overcome this limitation by learning to predict the 3D pose from available information.

Next, the ability to generalize to new scenes is important for deploying 3D human pose estimation in the wild. As explained in Section 2.5, existing datasets are limited to 'in-the-lab' conditions, making it difficult to train a network that directly predicts 3D pose from images. Hence, decoupling 2D pose estimation and 2D to 3D pose lifting allows the proposed framework to generalize to new scenes. In other words, the 2D pose estimation model can be trained on 'in-the-wild' datasets while the 2D to 3D pose lifter can be trained on synthetic 2D-3D pose pairs.

Finally, the compatibility with new camera setups is important for deploying 3D human pose estimation in the wild. Existing methods are limited to specific camera setups, making them difficult to deploy in new environments. To overcome this limitation, the proposed framework uses 3D ray representation as input, which are invariant to camera transformations. This allows the proposed framework to be compatible with new camera setups, making it more deployable in the wild. Next section discusses the 3D ray representation in more detail.

Table 3.1 Comparison of works based on input modality and various properties.

Input Modality	Paper	Recovery of occluded keypoints	New scenes	Compatibility to new camera setup
Image	Learn. tri. [3]	✓	×	×
	PPT [9]	✓	×	✓
2D Keypoints	Triangulation	×	✓	✓
	MPL [69]	✓	✓	×
3D Rays	**RMPL (ours)**	✓	✓	✓

3.3 3D Ray Representation of Poses

Training a network compatible with new camera setups is not a trivial task. We show in Section 6.6 that feeding the network directly with 2D keypoints (in pixel space) is not sufficient for training a network compatible to arbitrary camera setups. Moreover, we tried feeding the network with 2D keypoints along with camera calibration information (intrinsic and extrinsic). However, it seems that formulating the problem this way does not yield good results.

Therefore, we propose a novel 3D ray representation for alleviating the need for learning the camera calibration by the network. This ray representation consists of a vector indicating the direction of the ray and a point, *i.e.*, the camera location, on the ray. Let $P \in \mathbb{R}^{M\times 2}$ denote the M pairs of coordinates defining the location of the M keypoints in the pixel space. Intrinsic parameters of the camera are defined by K (3×3). Extrinsic parameters of the camera are defined by R (3×3) and T (3×1) containing the rotation matrix and the location of the camera, respectively. For converting the 2D pixel coordinates to 3D rays, we follow the steps below:

1. First, we convert the 2D pixel points into homogeneous coordinates by appending 1:

$$P_{hom} = [P \;\; 1]. \tag{3.1}$$

   This results in an $M \times 3$ matrix. Then, we transform it into normalized camera coordinates by applying the inverse of the intrisic matrix:

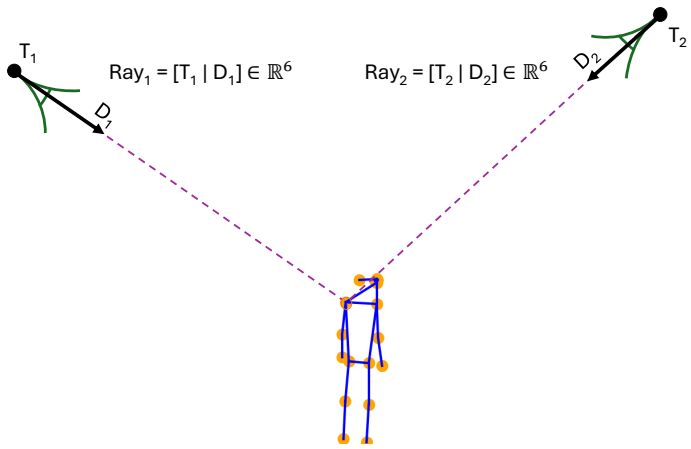


Fig. 3.2 3D Ray representation of poses. Representing the keypoints in the format of the 3D rays (instead of 2D coordinates) omits the requirement of learning the camera calibration by the network.

$$P_{norm} = K^{-1} P_{hom}^{T}. \tag{3.2}$$

This gives a $3 \times M$ matrix where each column represents a point in normalized camera coordinates.

2. We compute the direction of each ray originated from the camera center T.

$$D = R^{T} P_{norm}. \tag{3.3}$$

This gives a $3 \times M$ matrix where each column represents a direction corresponding to a keypoint.

3. Finally, we concatenate the directions and the camera locations to represent the 3D ray for each keypoint.

$$\mathcal{R} = [T \mid D^{T}]. \tag{3.4}$$

Fig. 3.2 illustrates an example of the 3D ray representation of keypoints. As shown in the figure, the rays are invariant to camera transformations,

making the network compatible with new camera setups.

4

Ray-based Multi-view 2D to 3D Pose Lifter (RMPL)

This chapter presents the Ray-based Multi-view 2D to 3D Pose Lifter (RMPL), a novel framework for 3D human pose estimation from multi-view images. Based on the insights presented in Chapter 3, RMPL takes 2D keypoints from multiple views from an off-the-shelf 2D pose detector and lifts them to 3D using a transformer-based architecture.

As shown in Fig. 4.1, RMPL is composed of two stages. First, the images from N different views are fed to an off-the-shelf monocular 2D pose estimator to get N skeletons made of M 2D keypoints. Second, the 2D poses are transformed to 3D rays using the provided camera calibration, and fed to a 3D pose lifter to get a 3D pose.

RMPL takes as input the 2D pose skeletons predicted by an off-the-shelf 2D pose estimator for N views. Let $P_i \in \mathbb{R}^{M\times 2}$ and $C_i \in \mathbb{R}^{M}$ denote the M pairs of coordinates defining the location and the confidence (as predicted by the off-the-shelf 2D pose estimator) of the M keypoints in the i^{th} view, respectively. RMPL takes $\{P_i\}_{i<N}$ as input and predicts the 3D pose skeleton $Q \in \mathbb{R}^{M\times 3}$ in the world coordinate system. RMPL includes three main components, namely the Keypoint to Ray Converter, the View Fusion Transformer (*VFT*), and the Pose Fusion Transformer (*PFT*), defined as follows.

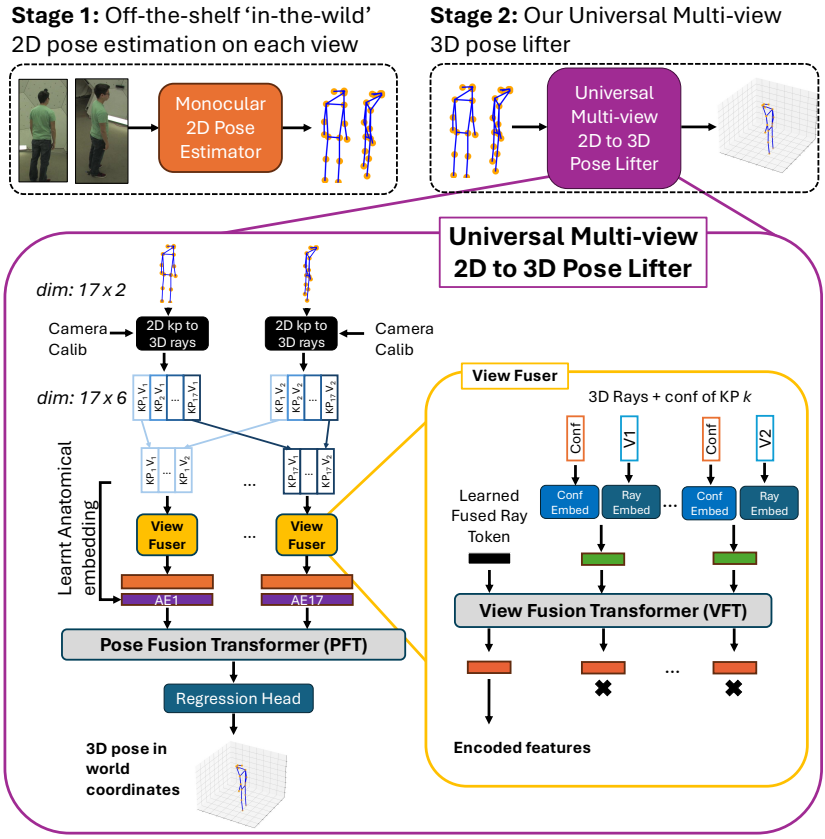


Fig. 4.1 RMPL takes 3D rays of all keypoints from all views, fuses the rays associated to a given keypoint and then jointly considers the whole set of keypoints using a transformer network.

4.1 Keypoint to Ray Converter

This module takes as input the keypoint coordinates $P_i \in \mathbb{R}^{M\times 2}$ and the camera calibration matrices K, R, and T and returns 3D rays $\mathcal{R}_i \in \mathbb{R}^{M\times 6}$ following the steps in Section 3.3.

As shown in Fig. 3.2, thanks to the 3D ray representation of 2D keypoints in RMPL, the 3D pose estimation is invariant to the camera calibration parameters. This is a key advantage of RMPL, as it allows the model to

be trained on synthetic data with arbitrary camera calibration parameters and then be used in real-world scenarios without the need for retraining.

4.2 Keypoint-level View Fusion Transformer (VFT)

VFT aggregates the information associated to a given keypoint (from all views) independently of the other keypoints. It is identical for all keypoints and encodes the keypoints in a format suited to the subsequent 3D pose estimation module. Let $\mathcal{R}_j \in \mathbb{R}^{N\times 6}$ denote the N 3D encoded rays from all views of the j^{th} keypoint. Note the conversion process from 2D keypoints to 3D rays in Fig. 4.1.

VFT takes $\mathcal{R}_j \in \mathbb{R}^{N\times 6}$ as input and returns a D-dimensional hidden embedding for each view. This is done as follows; first, a linear projection is applied to project each encoded ray to a $D/2$-dimensional vector. Another linear projection is applied to project confidence C_j to a $D/2$-dimensional vector. These two vectors are then concatenated to shape a D-dimensional vector, resulting into $X_j(i) \in \mathbb{R}^D$ for the i^{th} view.

Inspired by the usage of the transformers in classification tasks [70], we consider a learnable ray token $X_j(0) \in \mathbb{R}^D$ called fusion token which is expected to contain the fused features at the output of the transformer. It is worth noting that this approach gives RMPL the ability to accept any number of views without the need for retraining. This is because the architecture, in no way, is dependent of the number of views.

Next, the N ray tokens associated to a keypoint are fed to an encoder transformer that returns N vectors $X_j^V(i) \in \mathbb{R}^D$, with $0 \leq i \leq N$, for keypoint j. We adopt the same architecture as in [52] for the encoder transformer layers. It relies on multi-headed self attention (MHSA) and multi-layer perceptron (MLP). The transformer network has L layers and each layer has H heads [71].

Finally, the fusion token $X_j^V(0)$ for j^{th} keypoint is picked as the fused features of the corresponding keypoint resulting in $Y(j) \in \mathbb{R}^D$ for keypoint j.

4.3 Pose Fusion Transformer (PFT)

PFT takes the tokens fused by VFT as $Y(j) \in \mathbb{R}^D$ for each keypoint j, with $1 \leq j \leq M$. An anatomical encoding $AE(j) \in \mathbb{R}^D$, learned for each keypoint j, is first added to each $Y(j)$. This is expected to make the transformer aware of the the joints type information [71], resulting in a keypoint token $Y'(j) = Y(j) + AE(j)$ for the j^{th} keypoint. Next, the keypoint tokens are fed to an encoder transformer, which returns a fused embedding $Y^P(j) \in \mathbb{R}^D$. The transformer encoder in PFT follows the same scheme as that of VFT. Eventually, Y^P is passed to a regression head to make the final output, i.e., the 3D pose skeleton $Q \in \mathbb{R}^{M \times 3}$. The regression head is composed of a 1-layer MLP to predict the 3D pose in the world coordinates.

4.4 Loss Function

To train RMPL, the output is directly compared to the ground truth 3D pose using Mean Per Joint Position Error (MPJPE). Eq. (4.1) shows how MPJPE is calculated, with Q and $\hat{Q}$ denoting the predicted and the ground truth 3D pose, respectively.

$$\text{MPJPE} = \frac{1}{M} \sum_{j=1}^{M} \|Q_j - \hat{Q}_j{}^2\|. \tag{4.1}$$

5

Synthetic Dataset Generation

This chapter describes the generation of a synthetic dataset of 2D-3D pose pairs for training the 2D to 3D pose lifter. The most natural way to generate the 2D-3D pairs required to train RMPL is to randomly position a 3D skeleton in arbitrary positions within the observed 3D space and a number of cameras in the same scene, project this 3D skeleton into a view of interest, thereby generating the corresponding 2D skeleton.

This strategy, however, suffers from two important drawbacks. First, it appears that the 3D skeleton datasets adopt distinct rules to define the keypoints. Most often, those definitions are not compatible with the ones adopted by the off-the-shelf 2D pose estimation model used to process 2D images at inference, making the 2D-3D associations considered at training incompatible with the 2D pose manipulated at inference. Second, this natural 3D skeleton projection strategy is unable to mimic the noise and errors affecting 2D pose estimation at inference.

5.1 Mesh-based Data Generation

To alleviate issues mentioned above, following our preliminary work in [69] and [72], instead of considering 3D skeleton and their 2D projections

to train our RMPL, we propose to generate the 2D-3D pairs of skeletons from meshes.

Therefore, we make use of AMASS [68], an archive providing a variety of human shapes collected from many motion capture datasets and represented with standard 3D mesh models, convertible to SMPL vertices [40] that can be transferred to keypoints by regression [44]. Please refer to Section 2.5.3 for more details about AMASS.

Considering a dataset of 3D meshes offers two valuable properties. First, the 2D pose associated with a 3D mesh can be created by applying the off-the-shelf 2D pose estimation model used at inference on synthetic 2D renderings of the 3D mesh. This makes the 2D-to-3D pose lifter robust to noise induced by the 2D pose estimator. Second, the regressor used to convert the 3D mesh into a 3D skeleton can be designed to be compatible with the definition of keypoints adopted by the 2D pose estimator.

5.2 Mesh-based Human Pose Dataset Generator (MHP)

Fig. 5.1 shows the pipeline of our proposed Mesh-based Human Pose Dataset Generator (MHP) for 2D-3D pairs of human skeletons generation. It depicts how the 3D poses are generated by regressing each AMASS 3D mesh to an M-joint skeleton. Given the calibration parameter of a camera, the 3D mesh is rendered in the image space using the open-source software **Body Visualizer** [41] and **Trimesh** [46]. Then the rendered images are passed through the 2D pose estimator, *i.e.* the same as the one used at inference, to obtain the M-joint 2D poses. Finally, the predicted 2D poses from different views are paired with the regressed 3D keypoints to be used during training. An overview of this process is shown in Fig. 5.1.

It is worth mentioning that the camera locations are picked randomly within a pre-defined range for every 3D pose to increase the diversity of the projective observations. Thanks to utilizing the 3D ray representation of the 2D keypoints in RMPL, the keypoints' representations are independent of the camera calibration (see Fig. 3.2) resulting in a universal trained model, *i.e.* valid for arbitrary camera parameters (within the range used at training). This is in contrast with our preliminary work in [69] that has to be trained a specific model for each acquisition setup.

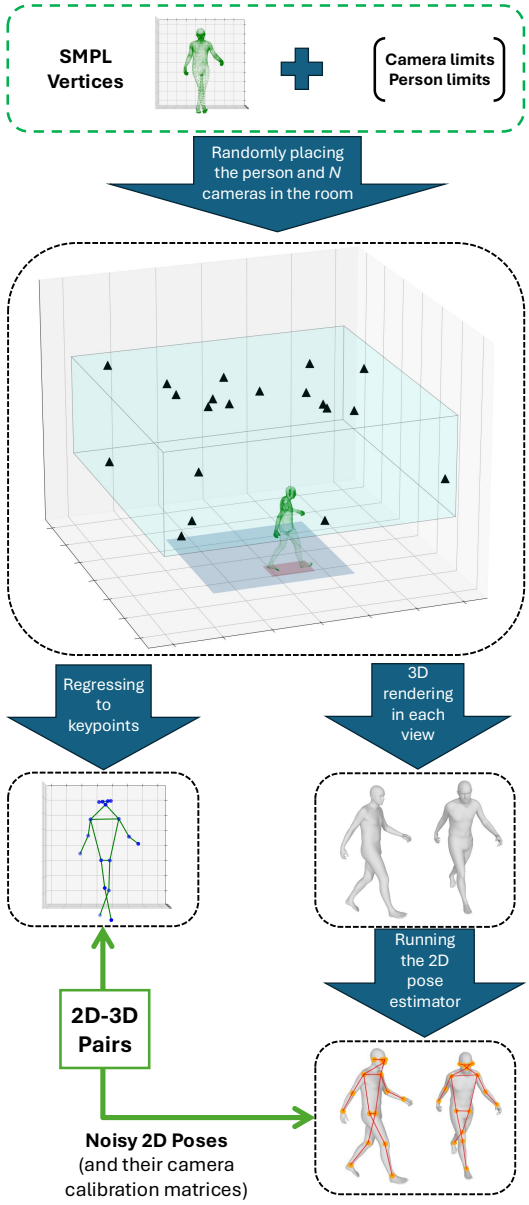


Fig. 5.1 Our **Mesh-based Human Pose Dataset Generator (MHP)** takes 3D mesh vertices and the limits of camera and person's location as inputs. Note that the triangles represent the location of the cameras, the light blue cube illustrates the camera limits, and the blue surface depicts the person's limits in the room.

6

Experimental Validation

This chapter presents the experimental validation of the proposed Ray-based Multi-view 2D to 3D Pose Lifter (RMPL). It first describes the metrics used to assess the performance of RMPL and the datasets used for its evaluation. Then, it presents the implementation details of RMPL and the training procedure. Finally, it reports the results of the experiments and compares RMPL with state-of-the-art methods.

6.1 Evaluation Metric

The 3D pose is evaluated by MPJPE (see Eq. (4.1)), computed in world coordinates in *mm*, between the ground truth 3D pose and the predicted 3D pose.

6.2 Datasets

We evaluate our RMPL and alternative methods on three commonly used 3D human pose datasets, Human3.6M [6], CMU Panoptic [1], and RICH [65]. In all our experiments, no image or 3D pose of the evaluation dataset is used to design or train the tested method.

6.2.1 Human3.6M

It is the most widely used public dataset for single-person 3D HPE (see Section 2.5.1). Prior works [9,44,52], utilized all the 15 actions for training their models on five subjects (S1, S5, S6, S7, and S8) and tested them on two subjects (S9 and S11). We consider the same two subjects for testing our network, but do not rely on Human3.6M images or 3D poses to train our RMPL model. For evaluating our RMPL on Human3.6M, we generate the RMPL training 2D-3D pose dataset using our mesh-based human pose (MHP) generator (see Chapter 5).

It is worth noting that the definition of keypoints in the Human3.6M dataset differs from the one introduced by COCO [16] and adopted by the off-the-shelf 2D pose estimator used in the study. As in [69], we have improved the consistency between the two formats by averaging the location of several COCO keypoints. For instance, averaging the *ears* and *eyes* locations from the COCO format results in a keypoint that closely matches the *head* keypoint in the Human3.6M format. Similarly, averaging the *hips* in COCO provides the *pelvis* in Human3.6M, and the mean of the *neck* and the newly created *pelvis* results in a *torso* keypoint. Fig. 6.1 illustrates this process. The set of keypoints that adopt consistent definitions in both formats is denoted KP* and includes the knees, the ankles, the shoulders, the elbows, and the wrists.

6.2.2 CMU Panoptic

This dataset is a more recent multi-view public dataset made available by the Carnegie Mellon University (see Section 2.5.1). Following previous works [3,73], we use the 17-keypoint subset (from the 19-joint annotation format) that are in line with the popular COCO format [16].

For evaluating on CMU, we generate the RMPL training 3D-2D pose dataset utilizing our MHP with AMASS meshes. Note that the CMU motion capture that is part of the AMASS collection has not been included in any of our training datasets. Finally, we test our RMPL on the test split recommended in [73], where only single-person scenes are selected.

6.2.3 RICH

As mentioned in Section 2.5.1, the RICH dataset is a large-scale multi-view dataset that includes 3D human poses in the form of SMPL-X meshes. We

extract 3D ground truth keypoints from the provided SMPL-X [41] meshes for our evaluation.

For evaluating on RICH, we generate the RMPL training 3D-2D pose dataset utilizing our MHP with AMASS meshes.

6.3 Implementation Details

For predicting the 2D pose in the image space, we use the open-source pose estimator framework **MMPOSE** [34]. It involves the popular HRNet [30] architecture, pretrained on COCO [16]. The average precision (AP) [16] of the MMPOSE 2D output on AMASS rendered images is 80.0%, 93.0%, and 84.7% when applied with person and camera displacements corresponding to the ones observed in the Human3.6M, CMU, and RICH datasets, respectively. Note that AP is computed when Object Keypoint Similarity (OKS) [16] threshold is set to 0.75 [1]. This shows that the chosen off-the-shelf 2D pose estimator is able to effectively handle 3D rendered human mesh images.

We set M, *i.e.* the number of joints, to 17. The hidden embeddings dimension D is set to 256 in our RMPL. Both VFT and PFT have the same configuration parameters. The number of encoder heads H is set to 8, and the number of encoder layers L is set to 12 for all the experiments. This choice of design follows that of [52]. During RMPL training, the input batch size is set to 32, and the network is optimized using the Adam optimizer [74] with MPJPE loss, for 20 epochs. The learning rate is initialized at 0.0001, and decays at the 10th and 15th epochs with a decreasing factor of 0.1.

6.4 Comparison to Multi-view Prior Works

Since our contribution primarily aims to enable 3D pose estimation in arbitrary observation conditions, a fair positioning should compare our RMPL to works that are either not trained (typically because they use triangulation on top of 2D pose estimation) or trained on other datasets than the one used at testing. Trained alternatives with publicly available code and

[1] Please refer to [16] for more information about OKS and AP.

trained weights are rare among previous works. We have identified three methods, namely Learnable-triangulation [3], TransFusion [8], and PPT [9].

Learnable-triangulation [3] aggregates features extracted from the image spaces by unprojecting them into a common voxel space. The publicly available data only provides the trained weights on Human3.6M; thus we test their network on CMU dataset.

TransFusion [8] trains a transformer-based network that, by globally attending features in the image space, fuses the information from different views. The trained weights are only provided on Human3.6M so we test their network on CMU and RICH where it completely failed to output any meaningful output. We believe this is caused by the global attention mechanism making it dependent of the scenery features.

Finally, PPT [9] follows the same approach as TransFusion but, instead of considering a global attention process in the fusion process, it prunes the unimportant tokens. Trained weights are only provided on Human3.6M. However, since PPT is more recent and provides more accurate results than TransFusion [9], we follow their guidelines to train their network on CMU dataset and also test on Human3.6M.

In addition, we compare our RMPL to the preliminary MPL [69] in two different variants. First, we keep the MPL architecture, feed it with 2D keypoints in pixel space, and second, we change the input tokens to 3D rays (same way as in RMPL) and feed it to the MPL architecture. The reason behind this comparison is to evaluate the impact of the changes affecting the RMPL architecture compared to the MPL one, beyond the input format, and to show the importance of the proposed architecture in the accuracy of the 3D pose estimation. We also compare our RMPL to a 3D pose lifter implemented based on a Fully Connected network, following the guidelines in [61] (with 3D ray formatting of the inputs considered for RMPL), instead of our transformer-based solution.

6.4.1 Human3.6M

Table 6.1 shows the 3D HPE results on the test set of Human3.6M for each action. To evaluate the impact of the mismatch between the COCO and Human3.6M formats (see Section 2.5.1) on the evaluation metrics, the last column in Table 6.1 computes the MPJPE only on the keypoints whose definitions are consistent in both the 3D and 2D formats. This set of keypoints is denoted KP* and corresponds to knees, ankles, shoulders, elbows, and

wrists.

The results show that PPT [9] is unable to generalize well to unseen images. Despite a higher number of views, it falls short both compared to the triangulation baselines and to our RMPL. This explains why triangulation is the reference when 3D pose estimation has to be deployed 'in-the-wild'.

The results show that utilizing MPL and using image coordinate tokens as inputs fails compared to other methods. However when using the original MPL architecture coupled with the 3D ray tokens as input, we see an improvement of 37.6 *mm* in the MPJPE (KP*) on Human3.6M val set compared to triangulation.

This improvement becomes even more when utilizing RMPL architecture where it improves the MPJPE (KP*) by 57.9 *mm* on average when using two views. This corresponds to an MPJPE reduction of about 50.5%. This gain raises to 56.7% when accounting for keypoints for which the 3D ground truth definition is not consistent with the one adopted by the 2D pose estimator. This reveals the ability of our RMPL to predict 3D keypoints that (moderately) differ from the keypoints predicted by the off-the-shelf 2D pose estimator, thereby increasing the flexibility of our framework[2].

Fig. 6.2a shows the distribution of MPJPE (KP*) on Human3.6M val set for three different methods. This figure shows a big gap between the performance of RMPL and MPL methods and triangulation.

Moreover, Table 6.1 shows the importance of the utilization of keypoint confidence (from off-the-shelf 2D pose estimator) in our architecture. The results show that not using the confidence results in an increase of 16 *mm* in MPJPE (KP*).

Finally, Table 6.1 shows that replacing the RMPL transformer by a fully connected network (leading to a solution close to [61]) results in a significantly worse pose lifting.

[2]For example, 3D keypoints could be defined as internal body points, even if the associated 2D keypoints are defined based on visual, but 3D-inconsistent, cues in the projected views.

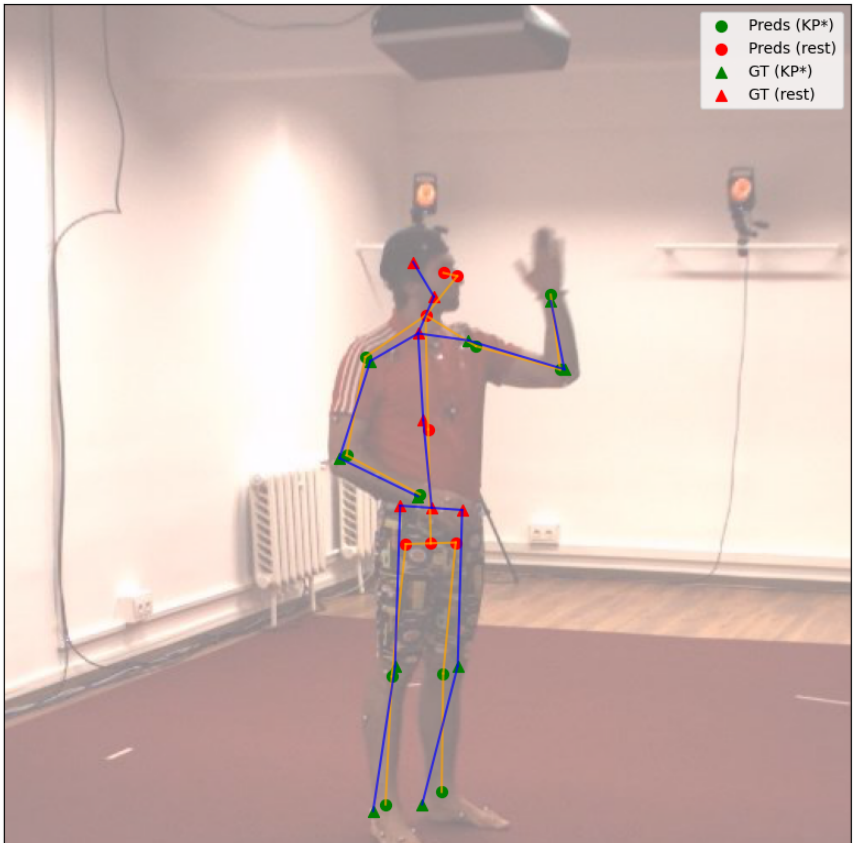


Fig. 6.1 Examples of inconsistency between Human3.6M and COCO format. The triangles depict the 3D ground truth keypoints projected to 2D space (Human3.6M format), and the circles depict the keypoints predicted by the off-the-shelf 2D pose estimator (COCO format). The green keypoints are the ones that are similarly defined in both formats, and the red ones correspond to the keypoints that have distinct definitions in the two formats. Note that, some of the predicted keypoints are estimated by averaging the locations of a subset of keypoints predicted by the 2D pose estimator (in COCO format), *e.g.*, the keypoint *torso* is located in the middle of *pelvis* and *neck*. We do not include these averaged keypoints in the set of keypoints KP*, corresponding to keypoints with similar definition, even if their location is reasonably similar to the one adopted in Human3.6M.

Table 6.1 3D MPJPE in *mm* on the Human3.6M test set for different methods. The last but one column averages the MPJPE on the multiple actions considered in the test set. The last column does the same but only considers the subset of keypoints KP* for which the 3D ground truth definition (adopted by Human3.6M) is consistent with the definition adopted by the MMPOSE 2D pose estimator. The numbers in bold indicate the lowest MPJPE with two views. Note that when written RCS, it means that random camera setup is used to create the training dataset (using MHP). It goes without saying that RMPL uses RCS by its definition.

Method	# Views	Dir.	Dis.	Eat.	Gre.	Phn.	Pht.	Pose	Pch.	Sit.	Sit. Down	Smo.	Wait	Walk Dog	Walk	Walk Two	Avg. (All KP)	Avg. (KP*)
PPT [9]	4	677.5	162.8	115.0	209.3	136.7	286.6	177.4	226.9	253.3	218.6	179.2	99.7	81.2	155.6	98.3	196.2	274.41
Triangulation	2	157.4	121.4	81.8	149.3	130.3	105.5	101.7	201.6	98.4	114.4	107.5	175.8	73.1	88.6	80.0	121.2	114.7
FullyConnected [61] + RCS + Rays	2	111.7	104.9	106.4	129.2	142.3	91.5	118.0	230.4	185.9	132.1	120.6	155.1	100.9	113.0	103.5	131.0	146.4
MPL [69] + RCS	2	431.8	374.8	377.4	449.1	403.5	402.2	356.3	475.3	331.7	390.3	421.6	439.3	381.6	368.8	411.5	400.7	437.5
MPL [69] + RCS + Rays	2	66.0	55.4	47.0	72.9	77.0	49.3	48.3	134.8	73.2	68.9	54.6	97.3	46.9	54.1	48.2	67.9	77.1
RMPL (w/o Conf)	2	50.3	52.1	55.4	51.8	64.9	46.7	55.6	76.2	79.6	60.2	59.8	54.1	53.5	55.5	55.1	58.6	64.9
RMPL	2	**48.6**	**47.9**	**45.7**	**46.6**	**58.3**	**46.9**	**49.4**	**67.1**	**67.3**	**54.5**	**51.8**	**50.4**	**48.1**	**50.0**	**48.0**	**52.5**	**56.8**

6.4.2 CMU Panoptic

We follow [10,75] and evaluate our RMPL on the CMU dataset based on a subset of HD cameras (3, 6, 12, 13, 23). Table 6.2 shows the MPJPE in *mm* for different methods. In all methods, none of the images or 3D poses of the CMU dataset have been used in the training.

The results show that the off-the-shelf pre-trained methods all fail to generalize to the images in the CMU test set, even when enjoying higher number of views.

Table 6.2 confirms all the observations of Table 6.1. Fig. 6.2b shows a clear improvement when RMPL is used comparing the MPJPE (KP*) on CMU.

Table 6.2 The comparison of 3D MPJPE on CMU Panoptic test set between different methods in *mm*. Note that the MPJPE (KP*) indicates that the calculations were done only considering the keypoints most aligned with COCO as a fair comparison. The numbers in bold indicate lowest MPJPE with two views. Note that when written RCS, it means that random camera setup is used to create the training dataset (using MHP). It goes without saying that RMPL uses RCS by its definition.

Method	# Views	↓ MPJPE (All KP)	↓ MPJPE (KP*)
Learn.-triang. [3]	5	-	130.9
PPT [9]	4	114.2	108.3
Triangulation	2	43.0	44.0
FullyConnected [61] + RCS + Rays	2	108.6	127.4
MPL [69] + RCS	2	274.9	278.8
MPL [69] + RCS + Rays	2	45.4	55.7
RMPL (w/o Conf)	2	36.6	41.1
RMPL	2	**30.8**	**35.0**

6.4.3 RICH

Table 6.3 shows the 3D HPE results on the RICH val set. We evaluate our RMPL on the RICH dataset by picking all the possible camera pairs of cameras in the scenes. Table 6.3 shows an improvement of 6.1 *mm* of MPJPE

(KP*).

When the displacement of the person with respect to the room center (used as origin of the referential manipulated by RMPL) gets large, it becomes helpful to translate the origin of the referential towards the human location before applying RMPL. The inverse translation is then applied to the pose predicted by RMPL. In practice, the translation vector is computed as a weighted average of the joints 3D coordinates obtained by triangulation. The weight associated to a keyppoint is defined as the minimum of the 2D confidence associated to this keypoint in the multiple views. We name this process **Recentering**.

Table 6.3 The comparison of 3D MPJPE on RICH val set between different methods in *mm*. Note that the MPJPE (KP*) indicates that the calculations were done only considering the keypoints most aligned with COCO as a fair comparison. The numbers in bold indicate lowest MPJPE with two views. Note that when written RCS, it means that random camera setup is used to create the training dataset (using MHP). It goes without saying that RMPL uses RCS by its definition.

Method	# Views	↓ MPJPE (All KP)	↓ MPJPE (KP*)
Triangulation	2	59.7	54.5
FullyConnected [61] + RCS + Rays	2	191.0	211.4
MPL [69] + RCS	2	386.5	413.7
MPL [69] + RCS + Rays	2	66.1	75.3
RMPL (w/o Conf)	2	113.9	111.7
RMPL	2	77.5	75.6
RMPL + Recentering (w/o Conf)	2	57.9	56.7
RMPL + Recentering	2	**51.3**	**48.4**

6.5 Comparison to Monocular Baselines

A straightforward approach to predict 3D pose from an arbitrary camera setup is to make use of an off-the-shelf monocular depth estimator coupled with the monocular 2D pose estimator. In this section, we consider this method as a baseline to compare our RMPL used in single-view fashion.

We have identified two state-of-the-art monocular depth estimators: Apple ml-depth-pro [76] and DepthAnything [77, 78]. For this baseline, we first separately feed each image to the 2D pose estimator and the depth estimator. Then by combining the outputs, we get a 3D pose in world coordinates. Table 6.4 compares different methods on the three datasets when only one view is used. It shows that even though RMPL is not using depth as input, it still results in a better MPJPE compared to the baselines considered. Extending the distance between the cameras and the human, as done when going from the CMU to the RICH dataset, obviously increases the estimation error.

Table 6.4 The comparison of 3D MPJPE (KP*) on different datasets between the proposed baseline composed of a monocular 2D HPE and a monocular depth estimator and our RMPL.

Method	# Views	↓ MPJPE (KP*) on		
		Human3.6M	CMU	RICH
2D HPE + ml-depth-pro [76]	1	506.8	215.9	815.5
2D HPE + DepthAnything2 [78]	1	397.5	248.6	1135.5
RMPL	1	**176.3**	**95.9**	**524.0**

6.6 Importance of the 3D Ray Representation

To show the importance of utilizing our proposed 3D ray representation, we consider three different scenarios where we change the input modality of the RMPL. The considered scenarios are when we feed the network with (i) 2D keypoints, (ii) 2D keypoints and camera calibration, and (iii) 3D ray. Table 6.5 shows that feeding the network with 3D rays improves the MPJPE (KP*) by 51.1%, 59.7%, and 88.1% on Human3.6M, CMU, and RICH, respectively, compared to when 2D keypoints and camera calibration are used directly as input.

Table 6.5 The importance of using our proposed 3D ray representation. The numbers are reported when two views are considered at testing.

Input Modality of **RMPL**	↓ MPJPE (KP*) on		
	Human3.6M	CMU	RICH
2D keypoints	695.7	269.0	428.8
2D keypoints + Camera Calibration	118.1	87.2	400.5
3D Ray	**56.8**	**35.0**	**48.4**

6.7 Relevance of Our MHP

To show the importance of utilizing our MHP to generate the RMPL training set, we consider 3 different scenarios for training our RMPL: A) training on the 3D and 2D-projected poses of Human3.6M (or CMU), B) regressing the 3D poses from AMASS meshes, and training based on their corresponding projected 2D poses, and C) adopting our proposed MHP, *i.e.* regressing the 3D pose from AMASS and generating noisy 2D by running the 2D pose estimator on 2D mesh rendering.

Table 6.6 compares the 3D MPJPE of each scenario, when testing on Human3.6M, CMU, and RICH from top to down. We also include the baseline, *i.e.* triangulation, and only consider the keypoints in KP*, *i.e.* the keypoints for which the COCO's format (same as the 2D pose estimator format in our study) is consistent with that of Human3.6M. The number of views for all scenarios is set to two, and the MPJPE is averaged on all possible pairs of cameras, selected among the ones that are commonly used in previous works, *i.e.* for Human3.6M: all four available cameras, for CMU: HD cameras (3, 6, 12, 13, 23), and for RICH: all available cameras.

By comparing the Scenario A and B, we observe a 53% drop in MPJPE on Human3.6M dataset showing a benefit of using AMASS over CMU. However, when comparing the same scenarios on CMU and RICH dataset, we observe a slight increase in MPJPE showing that using AMASS over Human3.6M has no benefit by itself. Interestingly, the use of our 2D-3D MHP pairs appears to significantly improve the MPJPE, up to 27.7% compared to pairs derived from the datasets ground truths, thereby revealing that MPH is effective in enriching the training set.

Table 6.6 The importance of MHP. MPJPE values are reported in *mm*, the number of views is fixed to two. The testing dataset is Human3.6M, CMU, and RICH for the tables from top to down, respectively.

On **Human3.6M**				
Method	Scenario	Training Dataset	MHP	↓ MPJPE
Triangulation	-	-	-	114.7
RMPL (ours)	A	CMU	✗	167.1
	B	AMASS	✗	78.6
	C	AMASS	✓	**56.8**

On **CMU**				
Method	Scenario	Training Dataset	MHP	↓ MPJPE
Triangulation	-	-	-	44.0
RMPL (ours)	A	Human3.6M	✗	43.1
	B	AMASS	✗	44.4
	C	AMASS	✓	**35.0**

On **RICH**				
Method	Scenario	Training Dataset	MHP	↓ MPJPE
Triangulation	-	-	-	54.5
RMPL (ours)	A	Human3.6M	✗	56.9
	B	AMASS	✗	57.6
	C	AMASS	✓	**48.4**

6.8 Inference speed

To figure out whether our RMPL can be a viable alternative to triangulation, it is worth considering its run time. Table 6.7 shows the inference speed of RMPL in frames/second where each frame is composed of all the input camera views of a single person. The numbers are measured on an NVidia A10 with a 32-core Intel Xeon Gold 6346 CPU running at 3.10GHz. As shown in the table, RMPL is able to run in real-time even when 5 views are used with the minimum batch size. Note that, while increasing the batch size improves the speed, it increases the delay to the output stream.

Table 6.7 The inference speed (frames/second) of RMPL for different number of views and batch sizes.

# Views	Batch Size			
	1	2	4	8
2	96.5	190.0	370.0	679.1
3	95.1	190.0	368.9	699.2
4	92.2	181.0	344.8	669.8
5	89.3	162.4	292.0	467.6

6.9 Complexity

Table 6.8 shows the number of parameters of our 3D HPE. The number of parameters of the 2D HPE is taken from [30] as we used HRNet-W32 as the 2D pose estimator. The number of parameters of the RMPL is calculated by counting the total number of parameters when two views are used in the transformer network.

6.10 Execution Time of Data Preparation and Training

RMPL needs to be trained with data prepared by MHP. For the experiments in this paper, we make use of 128,109 meshes from AMASS. It ap-

Table 6.8 The number of parameters of our 3D HPE. The number of parameters of the 2D HPE is taken from [30]. The number of parameters of the RMPL is calculated by counting the number of parameters in the transformer network.

Model	Stage	# parameters
HRNet-W32 [30]	2D HPE	28.5M
RMPL	2D to 3D pose lifter	12.7M
Total	-	41.2M

proximately takes 21 hours to run MHP on 20 random cameras on 4 NVidia A10 with 32-core Intel Xeon Gold 6346 CPU running at 3.10GHz.

The RMPL training time depends on the number of layers and heads used in the transformer network. For a network for which the maximum number of views is set to 5, with the configuration mentioned in Section 6.3, it takes about 44 minutes to train RMPL on data prepared by MHP. The training measurements are done on the same device using only 1 GPU.

6.11 Robustness and Occlusion

6.11.1 Occlusion

In order to measure the robustness of RMPL to occlusions, we randomly occlude a number of keypoints from the input image during testing on Human3.6M and CMU datasets. This is done by randomly giving a probability between 0 and 1 to each keypoint to be occluded. If a keypoint is occluded, the area around it is covered with a black square of size 50 pixels. Fig. 6.3 shows the MPJPE on Human3.6M and CMU datasets when occlusion is added to the input image. The x-axis shows the probability of a keypoint being occluded, and the y-axis shows the MPJPE. The blue line shows the performance of RMPL, while the orange line shows the performance of triangulation. The results show that RMPL is more robust to occlusions than triangulation.

6.11.2 Random Keypoint Removal During Training

In order to make RMPL more robust to occlusions, we randomly remove a certain number of keypoints from the input 2D pose during training. This is done by randomly selecting a number of keypoints to be removed from the input 2D pose. Each keypoint is removed with a probability of 0.2. Table 6.9 shows the effect of this random keypoint removal on the MPJPE (KP*) on the three datasets.

Table 6.9 The effect of random keypoint removal during the training of RMPL. MPJPE values are reported in *mm*, the number of views is set to two.

RMPL training	↓ MPJPE (KP*) on		
	Human3.6M	CMU	RICH
Without random kp removal	66.1	35.7	50.8
With random kp removal	56.8	35.0	48.4

6.11.3 Robustness to Camera Calibration Errors

In order to assess the robustness of RMPL to camera calibration errors, we add a Gaussian noise to the camera extrinsic parameters (rotation and translation) during testing. The noise is sampled from a Gaussian distribution with a mean of 0 and a standard deviation of 2 degrees for rotation and 20mm for translation. Table 6.10 shows the effect of this noise on the MPJPE on the three datasets. In this table, we also show the performance of RMPL when trained with camera calibration noise and without it. The results show that RMPL is more robust to camera calibration errors when trained with noise.

It is worth noting that if we train RMPL with camera calibration noise and test it on datasets with no noise, its MPJPE is slightly worse than when trained without noise. The MPJPE on Human3.6M, CMU, and RICH increases by 38.1, 20.6, and 39 *mm*, respectively.

Table 6.10 The effect of camera calibration error on the RMPL. The camera calibration noise is synthetically added to the camera extrinsic parameters during testing. MPJPE values are reported in *mm*, the number of views is set to two.

Method	↓ MPJPE on		
	Human3.6M	CMU	RICH
Triangulation	19281.1	186.9	424.1
RMPL trained w/o camera calibration noise	221.9	103.3	193.8
RMPL trained with camera calibration noise	141.7	83.6	148.5

6.12 Qualitative Results

Fig. 6.6 shows some qualitative results comparing our RMPL with triangulation and our preliminary MPL coupled with random camera setup (RCS) and ray representation technique when methods are tested on CMU.

6.13 Effect of Max Number of Views

As mentioned in Chapter 4, our RMPL is compatible to any number of views as input thanks to its transformer view fusion scheme. However, during the training process, we have to set a maximum number of views. In other words, for each batch, we fix a number of views between 2 and a number associated to the maximum number of views. To study the effect of this parameter, we consider changing this parameter while testing it on 2 views from each dataset. Table 6.11 shows the change in MPJPE (KP*) on the 3 different test sets for different values for maximum number of views used during training.

This table shows that this parameter has a negligible impact on the accuracy of the RMPL when the number of testing views is set to two.

Table 6.11 Variation of MPJPE (KP*) when changing the maximum number of views used during training. Note that the test set used is always composed of two views.

Max # Views (During Training)	↓ MPJPE (KP*) on 2 Views		
	Human3.6M	CMU	RICH
2	76.0	35.0	46.7
3	61.0	34.7	46.7
4	62.2	34.2	48.1
5	56.8	35.0	48.4
6	61.3	35.2	49.2
7	59.7	34.2	49.7
8	57.4	35.9	50.0
9	57.4	36.1	51.1
10	57.7	37.8	50.9

6.14 Effect of Camera Geometry on MPJPE

To see the effect of the camera topology, we design an experiment where we position the cameras on a sphere with different angles between them. As the range of camera angles in Human3.6M, CMU, and RICH datasets is very limited, we run this experiment on the test set of AMASS dataset (which is composed of CMU MoCap data). It goes without saying that we do not use this test set in any of our training scenarios. Here, we run MHP on both training and test set of AMASS. For the test set, we consider a spherical setup (of radius 6 m) with the center placed in $[0, 0, 0]$. Then we place two cameras in a certain height with a specific angle. Next, we move one of them step by step and calculate the MPJPE on test set in the new camera setup. We repeat that for other heights. In other words, the two cameras are on a circle (at a given height on the sphere), and we change the angle between the cameras. The heights are set to 2.2, 3.2, and 4 m. Fig. 6.7 shows the camera setup on the sphere.

Then we train our RMPL the training set and test the trained model on the test set (each time choosing the two views corresponding to the angle under experiment). Fig. 6.8 shows the variation of MPJPE (mm) as a function of the angle (°) between the cameras. This figure shows that when

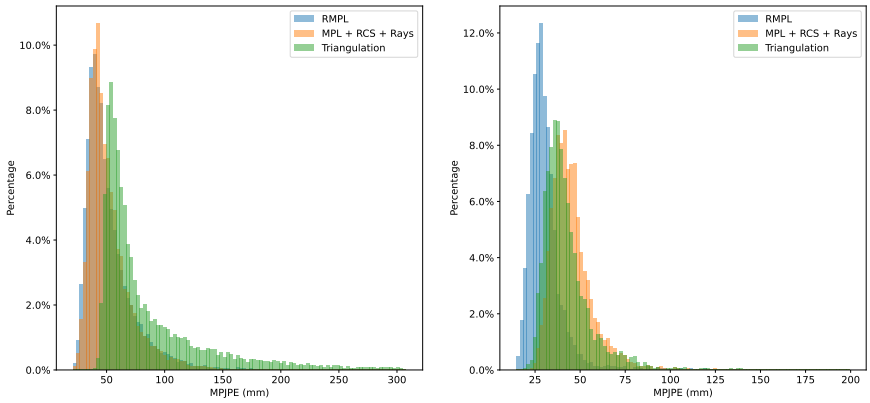
the angles get tighter or wider, the error goes higher.

6.15 Effect of Changing the 2D Pose Estimator

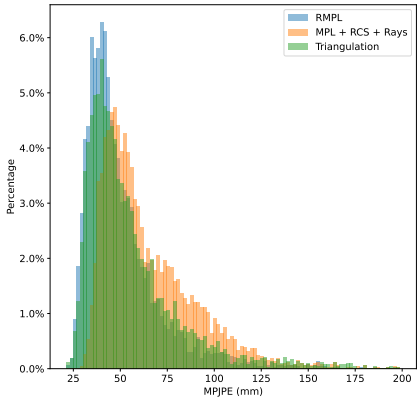
To see how much the noise bias of the 2D pose estimator affects the RMPL, we train RMPL with a different 2D pose estimator than the one used for testing. We use HRNet [30] from MMPOSE [34] for creating the training set using MHP and RTMPose [12] for during testing. Table 6.12 shows the effect of changing the 2D pose estimator on the MPJPE of RMPL. The table shows that the MPJPE increases when the 2D pose estimator used for testing is different from the one used for training. This shows the importance of the modeled by RMPL induced by the 2D pose estimator used for training.

Table 6.12 The effect of changing the 2D pose estimator on the MPJPE of RMPL. The numbers are reported when two views are considered at testing. The 2D pose estimator used for training is HRNet [30].

2D pose estimator for testing	↓ MPJPE on CMU
HRNet [30]	30.8
RTMPose [12]	77.1

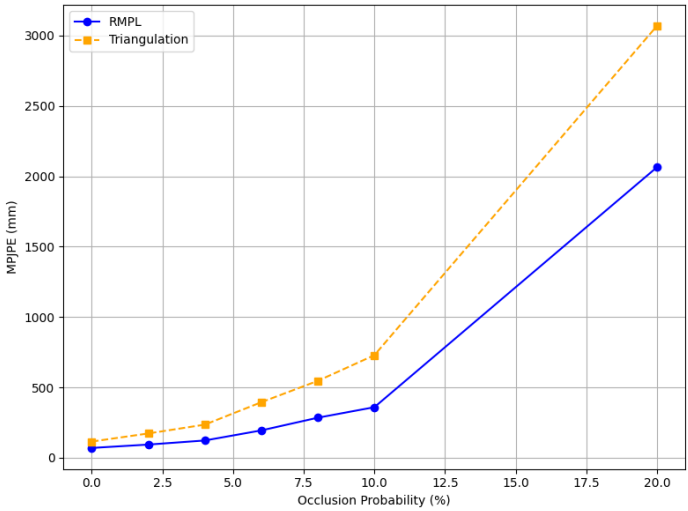


(a) On Human3.6M **(b)** On CMU

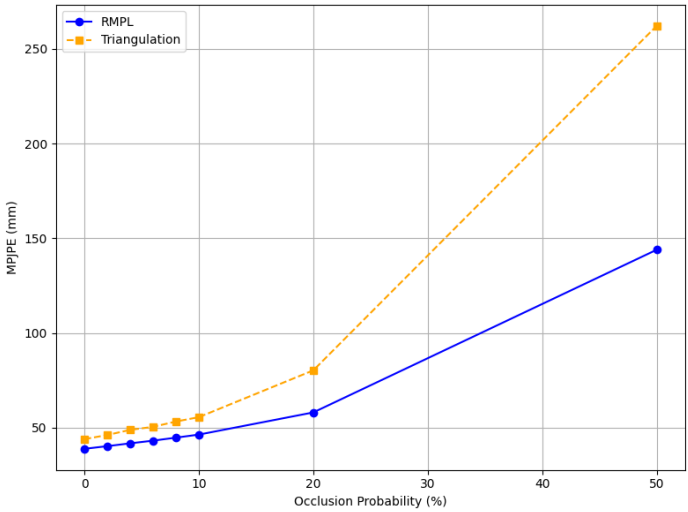


(c) On RICH

Fig. 6.2 Distribution of MPJPE (KP*) error on different datasets. Three different methods are compared on three different datasets. Note that the MPJPE (KP*) indicates that the calculations were done only considering the keypoints most aligned with COCO as a fair comparison. The numbers in bold indicate lowest MPJPE with two views. Note that when written RCS, it means that random camera setup is used to create the training dataset (using MHP). It goes without saying that RMPL uses RCS by its definition.



(a) On Human3.6M



(b) On CMU

Fig. 6.3 MPJPE comparison when occlusion added. The plot shows the MPJPE on Human3.6M and CMU datasets when occlusion is added to the input image. The x-axis shows the probability of a keypoint being occluded, and the y-axis shows the MPJPE. The blue line shows the performance of RMPL, while the orange line shows the performance of triangulation.

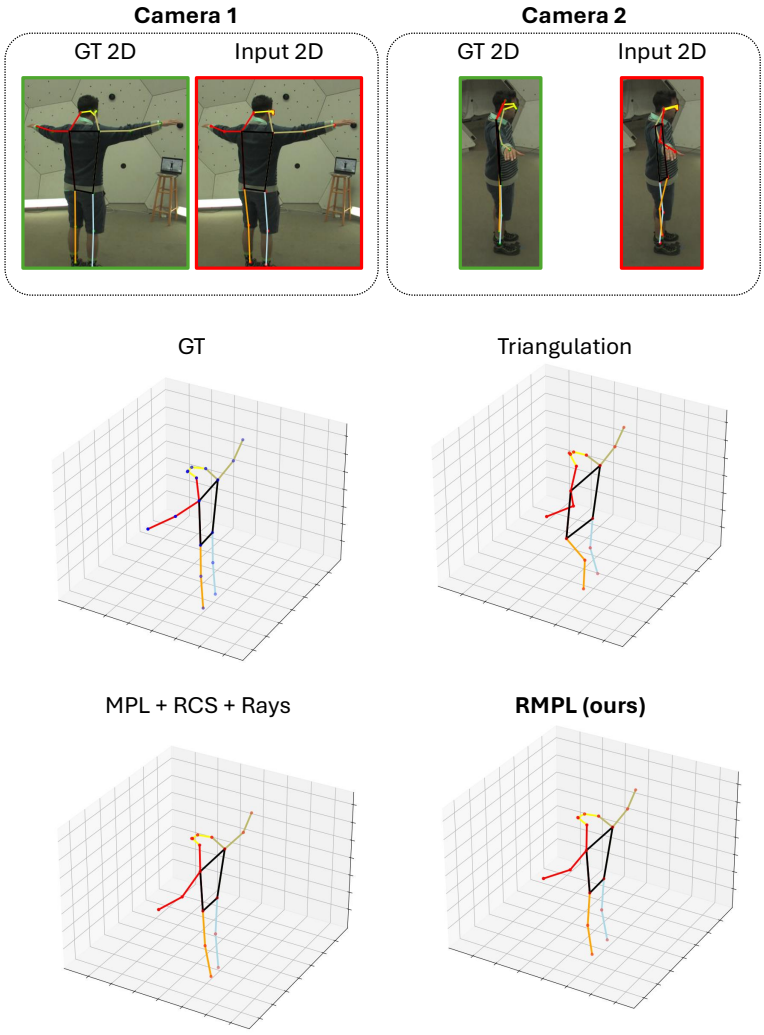


Fig. 6.4 Qualitative results. In the top row, the GT and **MMPOSE** 2D represent the ground truth taken from the CMU dataset and the predicted 2D pose from the off-the-shelf 2D pose estimator, respectively. In the middle row, GT shows the 3D ground truth from the CMU dataset, and the other pose corresponds to the output of triangulation when the top row is given as inputs. In the bottom row, the outputs of MPL coupled with random camera setup (RCS) and ray representation and RMPL are shown.

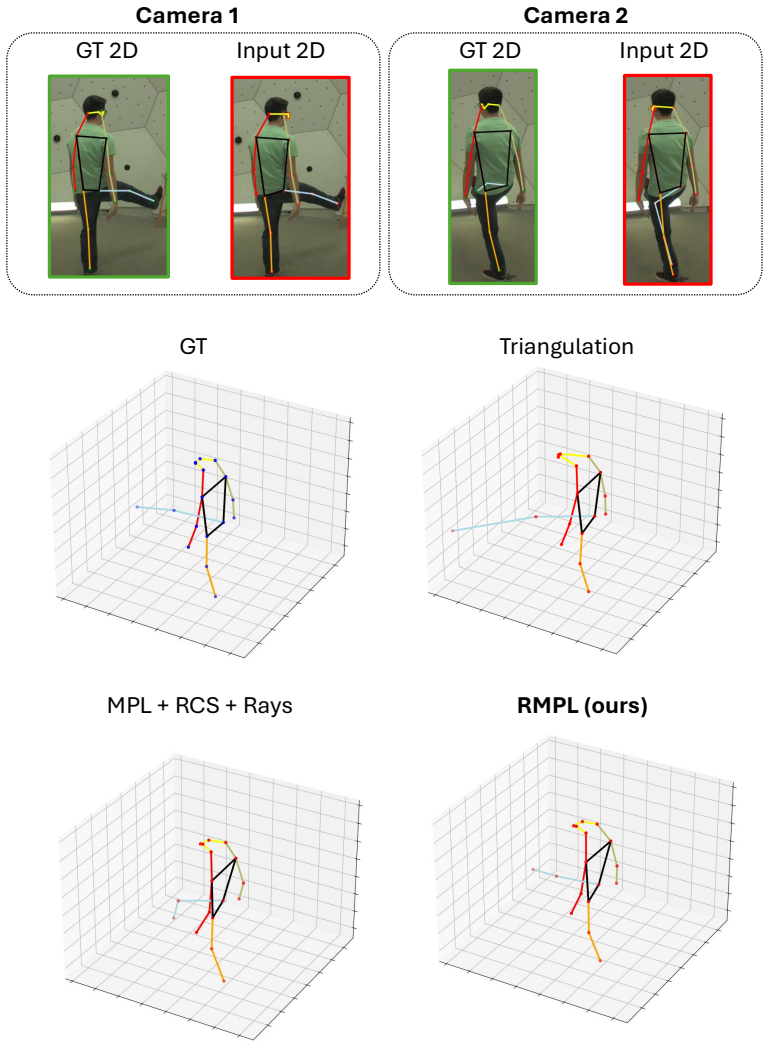


Fig. 6.5 Qualitative results (cont.)

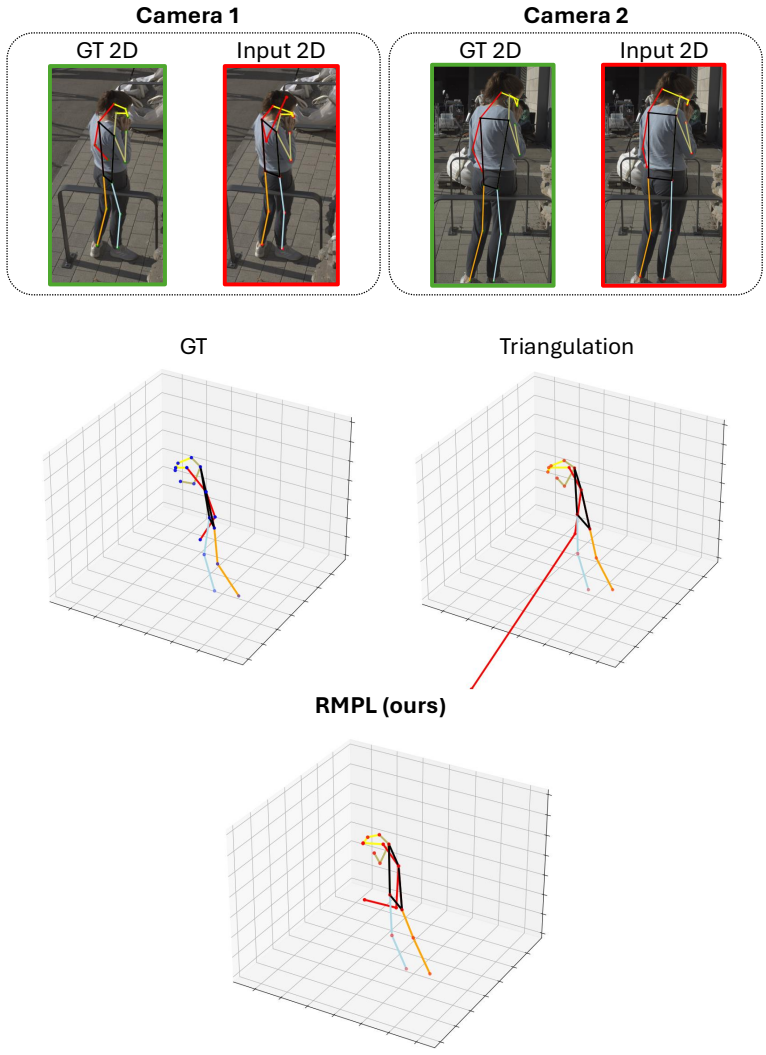


Fig. 6.6 Qualitative results (cont.)

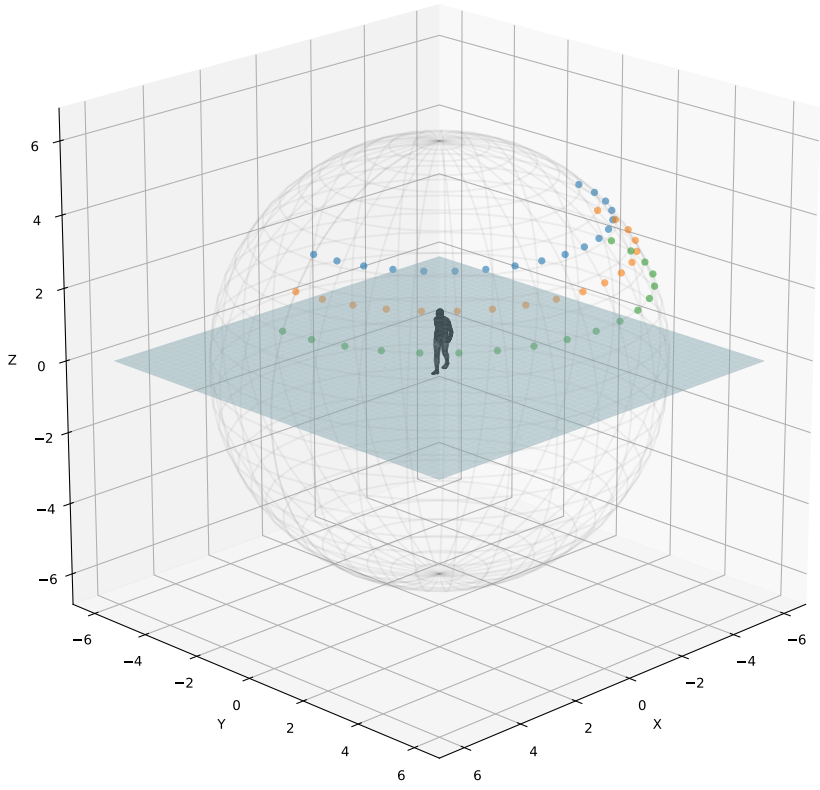


Fig. 6.7 The camera setup on a sphere with radius of 6 m and center of $[0, 0, 0]$. The heights are 2.2, 3.2, and 4 m.

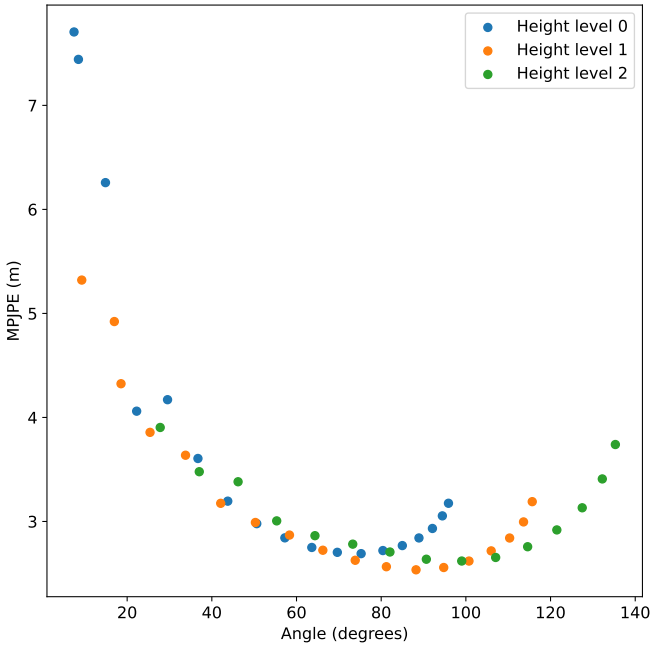


Fig. 6.8 The change in MPJPE when changing the composition of the cameras. The angles are calculated in a 2-view camera setup. The cameras are placed on a sphere with radius of 6 m and center of $[0, 0, 0]$. The heights are 2.2, 3.2, and 4 m.

7 Discussion

The development and validation of the Ray-based Multi-view 2D to 3D Pose Lifter (RMPL) presented in this thesis offer several insights into multi-view 3D human pose estimation, particularly in terms of its generalizability, efficiency, and robustness. This chapter reflects on the key findings derived from the experimental results and outlines the limitations of the proposed framework.

7.1 Insights from the Results

The experimental validation of RMPL across various datasets demonstrated the effectiveness of decoupling 2D pose estimation from 2D to 3D pose lifting. This decoupling approach offers several advantages: 1) Generalizability to in-the-wild conditions, 2) Independence from camera calibration, 3) Robustness to occlusions, 4) Scalability to any number of views, and 5) Reduction of 3D error. These insights are discussed in detail below.

7.1.1 Generalizability to In-the-wild Conditions

One of the core innovations of RMPL is its ability to generalize to real-world scenarios without retraining for specific acquisition conditions. By

training the pose lifter on synthetic 2D-3D pairs generated through the Mesh-based Human Pose Dataset Generator (MHP), the model effectively handles noise and variations encountered in real-world data. This significantly enhances the deployability of the framework in practical applications.

7.1.2 Independence from Camera Calibration

The use of 3D rays, instead of traditional 2D image coordinates, eliminates the need for detailed camera calibration. This is particularly important for real-world applications where the camera is not necessarily fixed in one position or has varying intrinsic parameters. The transformation of 2D keypoints into 3D rays removes the dependency on the camera's intrinsic and extrinsic parameters, thereby simplifying the deployment process.

7.1.3 Robustness to Occlusions

RMPL's transformer-based architecture, specifically the View Fusion Transformer (VFT) and the Pose Fusion Transformer (PFT), demonstrated an enhanced ability to handle occlusions. Unlike traditional triangulation methods that require keypoints to be visible in at least two views, RMPL can infer occluded keypoints by leveraging information from available views.

7.1.4 Scalability to Any Number of Views

A significant advantage of RMPL is its ability to handle any number of views during inference without retraining. This is achieved through the transformer-based architecture, which fuses information from multiple views in a scalable and efficient manner. The model's capacity to dynamically adjust to varying numbers of input views enhances its versatility in different deployment scenarios.

7.1.5 Reduction of 3D Error

The experimental results show a significant reduction in MPJPE compared to baseline methods. This is thanks to many factors, including the robustness of the transformer-based architecture, the compatibility with synthetic datasets, and the decoupling of 2D pose estimation from 2D to 3D pose

lifting. The proposed framework effectively addresses the limitations of existing methods and achieves more accurate and reliable 3D human pose estimations. The integration of confidence embeddings further contributes to the accuracy of the model.

7.2 Limitations of the Proposed Framework

Despite its advantages, the proposed RMPL framework has several limitations that need to be addressed. These limitations include reliance on off-the-shelf 2D pose estimators and the lack of temporal information in the current architecture.

7.2.1 Reliance on Off-the-shelf 2D Pose Estimators

The accuracy of RMPL still depends on the performance of the underlying 2D pose estimator. Even though the proposed framework is designed to be robust to noise and errors in 2D pose detections, the quality of the 2D pose predictions remains an important factor. While robust 'in-the-wild' 2D pose detectors are available, errors in 2D keypoint detection propagate to the 3D pose estimation stage. Addressing this dependency by integrating an end-to-end optimization framework could further improve accuracy.

7.2.2 Lack of Temporal Information

The current framework processes each frame independently, without leveraging temporal information from video sequences. Incorporating temporal consistency across frames could improve the stability and accuracy of the 3D pose predictions, particularly in scenarios where poses change dynamically over time. In addition, motion itself is a valuable source of information for 3D pose estimation and can help disambiguate poses that are otherwise similar in appearance. Future work could explore temporal models, such as temporal transformers, to embedd motion information into the RMPL framework. This could enhance the model's ability to capture dynamic movements and improve the overall performance of 3D pose estimation in video sequences.

8

Conclusion

This chapter concludes the thesis by summarizing the key contributions of the research and outlining potential directions for future work.

8.1 Summary

The Ray-based Multi-view 2D to 3D Pose Lifter (RMPL) presented in this thesis provides a novel and practical solution to the challenges posed by multi-view 3D human pose estimation. By decoupling the 2D pose estimation from the 2D to 3D pose lifting process, RMPL addresses several limitations inherent to previous methods. One of the most significant achievements of RMPL is its ability to handle arbitrary acquisition setups by leveraging synthetic 2D-3D pairs generated through a mesh-based human pose dataset generator called MHP. This flexibility allows the framework to achieve high generalizability and deployability in real-world scenarios, without requiring retraining for specific camera setups.

The experimental results confirmed that RMPL can reduce MPJPE significantly compared to baseline methods, even in cases where traditional approaches struggle due to occlusions or noisy 2D pose detections. Furthermore, the use of a transformer-based architecture not only enhances the framework's scalability to any number of views but also improves its

robustness against various data inconsistencies. Despite its promising performance, the challenges of reliance on off-the-shelf 2D pose estimators and lack of temporal information still remain. Addressing these challenges is key to unlocking the full potential of RMPL in practical applications.

The contributions of this research extend beyond academic boundaries, offering practical implications for industries that require accurate and efficient human pose estimation systems. These industries include sports analytics, security surveillance, healthcare monitoring, and immersive virtual environments. By providing a versatile and adaptable framework, this thesis lays the foundation for future innovations in these fields.

8.2 Perspectives

Looking ahead, several promising directions can further advance the work presented in this thesis. The exploration of temporal consistency in pose predictions is a promising direction. Video-based applications, such as motion capture for animation or behavioral analysis, require stable and coherent pose estimations over time. Incorporating temporal information into the RMPL framework could improve its performance in these applications by reducing jitter and ensuring smoother pose transitions. Moreover, the integration of motion information could help disambiguate similar poses, enhancing the model's ability to capture dynamic movements.

Next, exploring the integration of RMPL with other computer vision tasks, such as action recognition or object detection, could open up new possibilities for multi-modal applications. By combining pose estimation with other visual cues, the framework could provide richer insights into human behavior and interactions, enabling a wide range of innovative applications in fields such as robotics, augmented reality, and human-computer interaction.

The RMPL has the potential to be extended to new poses, body shapes, or even new skeleton definitions, *e.g.*animals or objects, by leveraging the mesh-based dataset generation approach. This would involve adapting the MHP to generate synthetic datasets for different body types or skeleton structures, allowing RMPL to generalize across various applications. Such an extension could significantly broaden the applicability of RMPL in fields such as biomechanics, wildlife monitoring, and robotics.

Furthermore, as shown in the experiments, the 2D pose estimators per-

form quite well on synthetic 3D mesh rendered images. However, a path to explore would be to make realistic images from those 3D rendered images to mimic even more the real-world conditions.

Additionally, the RMPL could be extended to support multi-person pose estimation, which is crucial for applications in crowded environments or social interactions. This would involve developing techniques to handle occlusions and interactions between multiple subjects, ensuring accurate pose estimations even in complex scenarios. This is, of course, possible with the current RMPL framework by matching the 2D detections from the same person across the different views using a Hungarian algorithm. However, as the occlusion patterns are different when multiple people are present, the RMPL could be trained to learn the occlusion patterns and adapt to them.

Finally, a potential avenue for future work is the exploration of camera calibration prediction implemented inside the RMPL framework. This would allow the model to adapt to different camera setups dynamically without the need for manual calibration. By incorporating camera calibration prediction, RMPL could further enhance its flexibility and usability in real-world applications, making it a more robust solution for diverse deployment scenarios.

In conclusion, the future of 3D human pose estimation holds significant promise. The RMPL framework, as presented in this thesis, provides a solid foundation for future advancements in the field. By addressing the challenges identified and exploring the proposed research directions, RMPL can continue to evolve and contribute to a wide range of applications in computer vision and beyond.

Acknowledgments

This work was funded by the FRIA/FNRS. Computational resources have been provided both by the super computing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

Bibliography

[1] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview system for social motion capture," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[3] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in *International Conference on Computer Vision (ICCV)*, 2019.

[4] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] Z. Wang, L. Chen, S. Rathore, D. Shin, and C. Fowlkes, "Geometric pose affordance: 3d human pose with scene constraints," *arXiv preprint arXiv:1905.07718*, 2019.

[6] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, p. 1325–1339, July 2014.

[7] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, and X. Xie, "Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 836–845, 2021.

[8] H. Ma, L. Chen, D. Kong, Z. Wang, X. Liu, H. Tang, X. Yan, Y. Xie, S.-Y. Lin, and X. Xie, "Transfusion: Cross-view fusion with transformer for 3d human pose estimation," in *British Machine Vision Conference*, 2021.

[9] H. Ma, Z. Wang, Y. Chen, D. Kong, L. Chen, X. Liu, X. Yan, H. Tang, and X. Xie, "Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation," in *ECCV*, 2022.

[10] T. Wang, J. Zhang, Y. Cai, S. Yan, and J. Feng, "Direct multi-view multi-person 3d human pose estimation," *Advances in Neural Information Processing Systems*, 2021.

[11] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, and W. Yang, "RTMO: Towards high-performance one-stage real-time multi-person pose estimation," 2023.

[12] T. Jiang, P. Lu, L. Zhang, N. Ma, R. Han, C. Lyu, Y. Li, and K. Chen, "Rtmpose: Real-time multi-person pose estimation based on mmpose," 2023.

[13] D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2637–2646, 2022.

[14] Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, and E. Zhou, "Tokenpose: Learning keypoint tokens for human pose estimation," in *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 11313–11322, 2021.

[15] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11977–11986, 2019.

[16] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.

[17] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4903–4911, 2017.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[19] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE international conference on computer vision*, pp. 2334–2343, 2017.

[20] S. Huang, M. Gong, and D. Tao, "A coarse-fine network for keypoint localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 3028–3037, 2017.

[21] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7103–7112, 2018.

[22] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 466–481, 2018.

[23] M. Kocabas, S. Karagoz, and E. Akbas, "Multiposenet: Fast multi-person pose estimation using pose residual network," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 417–433, 2018.

[24] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–286, 2018.

[25] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Advances in neural information processing systems*, pp. 2277–2287, 2017.

[26] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields,"

IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 1, pp. 172–186, 2019.

[27] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4929–4937, 2016.

[28] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pp. 34–50, Springer, 2016.

[29] S. Kreiss, L. Bertoni, and A. Alahi, "OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association," *arXiv preprint arXiv:2103.02440*, March 2021.

[30] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5693–5703, 2019.

[31] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, 2022.

[32] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

[33] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Semi-convolutional operators for instance segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 86–102, 2018.

[34] M. Contributors, "Openmmlab pose estimation toolbox and benchmark." `https://github.com/open-mmlab/mmpose`, 2020.

[35] W. Zhu, X. Ma, Z. Liu, L. Liu, W. Wu, and Y. Wang, "Motionbert: A unified perspective on learning human motion representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[36] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5137–5146, 2018.

[37] S. A. Ghasemzadeh, G. Van Zandycke, M. Istasse, N. Sayez, A. Moshtaghpour, and C. De Vleeschouwer, "Deepsportlab: a unified framework for ball detection, player instance segmentation and pose estimation in team sports scenes," 2021.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[39] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. C-22, no. 1, pp. 67–92, 1973.

[40] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, pp. 248:1–248:16, Oct. 2015.

[41] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pp. 561–578, Springer, 2016.

[43] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7122–7131, 2018.

[44] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3383–3393, 2021.

[45] Z. Li, J. Liu, Z. Zhang, S. Xu, and Y. Yan, "Cliff: Carrying location information in full frames into human pose and shape estimation," in *European Conference on Computer Vision*, pp. 590–606, Springer, 2022.

[46] Dawson-Haggerty et al., "trimesh."

[47] X. Sun, C. Li, and S. Lin, "An integral pose regression system for the eccv2018 posetrack challenge," *arXiv preprint arXiv:1809.06079*, 2018.

[48] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *Proceedings of the IEEE international conference on computer vision*, pp. 2640–2649, 2017.

[50] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, AAAI Press, 2018.

[51] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7753–7762, 2019.

[52] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d human pose estimation with spatial and temporal transformers," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[53] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan, "Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13232–13242, June 2022.

[54] W. Shan, Z. Liu, X. Zhang, Z. Wang, K. Han, S. Wang, S. Ma, and W. Gao, "Diffusion-based 3d human pose estimation with multi-hypothesis aggregation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14761–14771, October 2023.

[55] Q. Zhao, C. Zheng, M. Liu, P. Wang, and C. Chen, "Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8877–8886, June 2023.

[56] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu, "Epipolar transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7779–7788, 2020.

[57] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3d human pose estimation," in *International Conference on Computer Vision (ICCV)*, 2019.

[58] V. Srivastav, K. Chen, and N. Padoy, "Selfpose3d: Self-supervised multi-person multi-view 3d pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2502–2512, 2024.

[59] Z. Liao, J. Zhu, C. Wang, H. Hu, and S. L. Waslander, "Multiple view geometry transformers for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 708–717, 2024.

[60] K. Bartol, D. Bojanić, T. Petković, and T. Pribanić, "Generalizable human pose triangulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11028–11037, 2022.

[61] A. Kadkhodamohammadi and N. Padoy, "A generalizable approach for multi-view 3d human pose regression," *Machine Vision and Applications*, vol. 32, no. 1, p. 6, 2021.

[62] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3d human pose in the wild using imus and a moving camera," in *European Conference on Computer Vision (ECCV)*, sep 2018.

[63] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1, pp. 4–27, 2010.

[64] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total capture: 3d human pose estimation fusing video and inertial sensors," in *2017 British Machine Vision Conference (BMVC)*, 2017.

[65] C.-H. P. Huang, H. Yi, M. Höschle, M. Safroshkin, T. Alexiadis, S. Polikovsky, D. Scharstein, and M. J. Black, "Capturing and inferring dense full-body human-scene contact," in *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 13274–13285, June 2022.

[66] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, p. 139–144, Oct. 2020.

[67] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *CVPR*, 2017.

[68] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "AMASS: Archive of motion capture as surface shapes," in *International Conference on Computer Vision*, pp. 5442–5451, Oct. 2019.

[69] S. A. Ghasemzadeh, A. Alahi, and C. D. Vleeschouwer, "Mpl: Lifting 3d human pose from multi-view 2d poses," 2024.

[70] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

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

[72] A. Maiorca, S. A. Ghasemzadeh, T. Ravet, F. Cresson, T. Dutoit, and C. D. Vleeschouwer, "Self-avatar animation in virtual reality: Impact of motion signals artifacts on the full-body pose reconstruction," 2024.

[73] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10957–10966, 2018.

[74] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[75] H. Tu, C. Wang, and W. Zeng, "Voxelpose: Towards multi-camera 3d human pose estimation in wild environment," in *European Conference on Computer Vision (ECCV)*, 2020.

[76] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth pro: Sharp monocular metric depth in less than a second," *arXiv*, 2024.

[77] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *CVPR*, 2024.

[78] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv:2406.09414*, 2024.