UNIVERSITY OF CALIFORNIA,
IRVINE

# Robust Estimation of 3D Human Body Pose with Geometric Priors

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Zhe Wang

Dissertation Committee:
Professor Charless C. Fowlkes, Chair
Professor Xiaohui Xie
Professor Sameer Singh

2021

# DEDICATION

To my pain and joy in Irvine.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

First of all, Let me express my deepest thanks to my parents who support me from Childhood up till now. Without all of their support I cannot go this far.

I want to thank my academic advisor Professor Charless Fowlkes. He introduced me the way to 3D computer vision and taught me about traditional computer vision and the way to improve the quality of works. It is my great honor to work with him.

I am extreme grateful for my final defense committee member Professor Charless Fowlkes, Professor Xiaohui Xie, and Professor Sameer Singh; And my additional advancement committee member Professor Shuang Zhao and Professor Zyg Pizlo.

I would like to thank Qiqi to shine my life on the last several months of my Ph.D.

I would like to express my special thanks to my friend Deying Kong and KiKi, Zhanhang Liang and Junjie Shen, Yingtong Liu and TACO, their support during covid helped me to overcome loneliness and be more happy.

I would also like to extend my gratitude for the industry guidance from my mentors at Amazon: Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Liu and Joe Tighe, and at Adobe: Jimei Yang, Jianming Zhang, Ersin Yumer, Duygu Ceylan.

I am fortunate to work with the collaborators Junhao Zhang, Tianyu Luan, Haoyu Ma, Liangjian Chen, Xiaoyi Liu, Liyan Chen, Yu Qiao, Limin Wang, Xiaohui Xie, Shaurya Rathore, Daeyun Shin, Yali Wang, Zhipeng Zhou, they helped me to enrich my research experience during my Ph.D academic life.

Thanks for the helpful discussion and input from UCI Computational Vision Lab (Raúl Díaz , Golnaz Ghiasi, Phuc Nguyen, Bailey Kong, Samia Shafique, James Supancic, Zhile Ren). I would like to thank many friends, especially Zhengli, Qi, Yao, Di, Gufeng, Pan, Lan and Meng. I can hardly exhaust this list and hope my other friends can forgive me for not being able to mention their names here. It has been a great pleasure to have you along during these years.

I would like to thank NSF grants IIS-1813785, IIS-1618806, IIS-1253538, CNS-1730158, and a hardware gift from NVIDIA. I also thank Shu Kong and Minhaeng Lee for helpful discussion, John Crawford and Fabio Paolizzo for providing support on the motion capture studio, and all the UCI friends who contribute to the GPA dataset collection.

# VITA

## Zhe Wang

### EDUCATION

**Doctor of Philosophy in Computer Science** **2021**
University of California, Irvine *Irvine, California*

**Bachelor of Engineering in Digital Media Technology** **2014**
Beijing University of Posts and Telecommunications *Beijing*

### RESEARCH EXPERIENCE

**Graduate Research Assistant** **2016–2021**
University of California, Irvine *Irvine, California*

**Applied Scientist Intern** **2020–2020**
Amazon Science *Irvine, California*

**Research Intern** **2017–2017**
Adobe Research *San Jose, California*

**Research Assistant** **2014–2016**
MMLAB *Shenzhen and Hongkong*

### TEACHING EXPERIENCE

**Teaching Assistant** **2016–2017**
University of California, Irvine *Irvine, California*

## REFEREED CONFERENCE PUBLICATIONS During PhD

**SSCAP: Self-supervised Co-occurrence Action Parsing for Unsupervised Temporal Action Segmentation** **2022**
Z Wang, H Chen, X Li, C Liu, Y Xiong, J Tighe, C Fowlkes, WACV

**The Best of Both Worlds: Combining Model-based and Nonparametric Approaches for 3D Human Body Estimation** **2022**
Z Wang, J Yang, C Fowlkes, arxiv 1905.07718

**TransFusion: Cross-view Fusion with Transformer for 3D Human Pose Estimation** **2021**
H Ma, L Chen, D Kong, Z Wang, X Liu, H Tang, X Yan, Y Xie, S Lin, X Xie, BMVC

**PC-HMR: Pose Calibration for 3D Human Mesh Recovery from 2D Images/Videos** **2021**
T Luan, Y Wang, J Zhang, Z Wang, Z Zhou, Y Qiao, AAAI

**Predicting Camera Viewpoint Improves Cross-dataset Generalization for 3D Human Pose Estimation** **2020**
Z Wang, D Shin, C Fowlkes, ECCVW

**Geometric Pose Affordance: 3D Human Pose with Scene Constraints** **2019**
Z Wang, L Chen, S Rathore, D Shin, C Fowlkes, Arxiv

**Structured Triplet Learning with POS-tag Guided Attention for Visual Question Answering** **2018**
Z Wang, X Liu, L Chen, L Wang, Y Qiao, X Xie, C Fowlkes, WACV

**Towards Good Practices for Visual Question Answering** **2017**
Z Wang, X Liu, L Chen, L Wang, Y Qiao, X Xie, C Fowlkes, CVPRW

**Weakly Supervised PatchNets: Learning Aggregated Patch Descriptors for Scene Recognition** **2017**
Z Wang, L Wang, Y Wang, B Zhang, Y Qiao, C Fowlkes, CVPRW

## REFEREED JOURNAL PUBLICATIONS During PhD

**Learning Dynamical Human-Joint Affinity for 3D Pose Estimation in Videos** **2021**
J Zhang, Y Wang, Z Zhou, T Luan, Z Wang, Yu Qiao, TIP

# ABSTRACT OF THE DISSERTATION

Robust Estimation of 3D Human Body Pose with Geometric Priors

By

Zhe Wang

Doctor of Philosophy in Computer Science

University of California, Irvine, 2021

Professor Charless C. Fowlkes, Chair

Accurate estimation of 3D human pose/shape from a single image remains a challenging task under occlusion or domain shift. We try to solve this problem by investigating three different geometric priors: camera pose priors, scene geometry priors, and parametric body-model priors. The first part of the dissertation focuses on analyzing the difference in the popular 3d human pose datasets and proposes a plug-in camera pose module to improve cross-dataset generalization. In the second part, we evaluate the usefulness of scene geometry in helping improve 3d pose estimators. Finally, we build strong pose/shape estimators from a single image by leveraging the best of the statistical model-based methods and nonparametric methods.

# Chapter 1

# Introduction

## 1.1 Problem Definition and Motivation

We formulate the problem of human pose estimation as determing the 3D locations of human joints relative to root joint (usually pelvis of human). This is shown in Fig 1.1.

Accurate estimation 3D human pose from image data is a crucial task in computer vision as it enables lots of useful daily applications like telecommunications, humanoid robots, automatic grocery store, motion analysis, virtual try-on as in Fig 1.2.

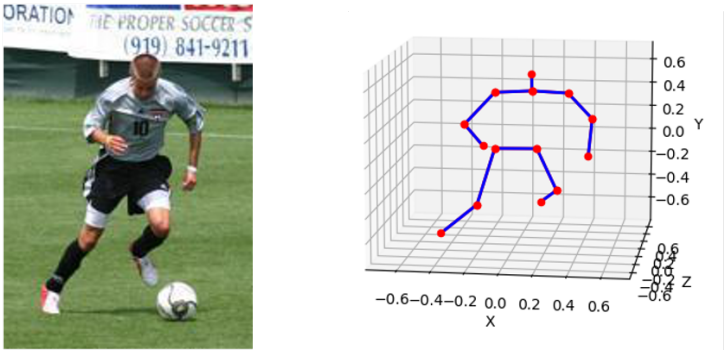



Figure 1.1: Problem definition for 3D human pose estimation.

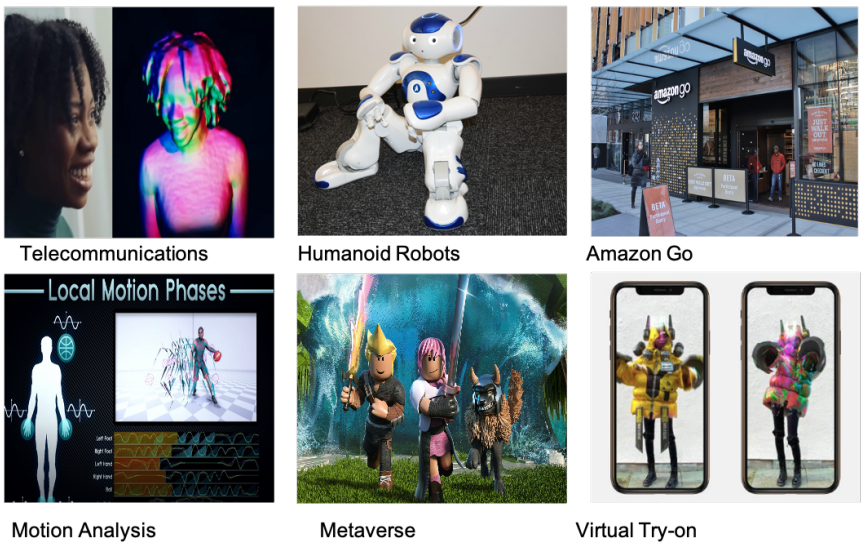



Figure 1.2: Illustration of 3D human pose applications related to daily life

However, this problem remains a challenging task as it faces several corners cases like arbitrary gesture, various skeleton size or rich background, self occlusion or object/scene occlusions as in Fig 1.3. Technically, the difficulties lie in several aspects: It is hard to obtain the groundtruth data as it requires 3D annotations; It is non-trival to represent the groundtruth as training target for the neural network; Additionally, developing or utlizing existing priors to get better pose estimation remain an open area; it is also demanding to integrate the training/inference of 3D human pose estimation with modern deep learning.

## 1.2 Dissertation Outline and Contributions

The general outline of the rest of the dissertation is as follows: Chapter 2 introduces some background knowledge, and related literature of 3D human pose estimation. Chapters 3-5 present the incorporation of camera pose priors, scene constraints, and parametric human model to have a more robust 3D pose estimators. To be more specific:

**Chapter 2**: We discuss how existing 3D human pose datasets have been collected and curated. In addition, we also discuss the design of networks and representations that incorporate


Figure 1.3: Illustration of 3D human pose applications related to daily life

general priors to handle 3D human pose estimation in deep learning era. Lastly, we discuss existing problems to be handled in 3D human pose estimation.

**Chapter 3**: We carry out a systematic study of the diversity and biases present in specific datasets and their effect on cross-dataset generalization across a compendium of 5 pose datasets. We specifically focus on systematic differences in the distribution of camera viewpoints relative to a body-centered coordinate frame. Based on this observation, we propose an auxiliary task of predicting the camera viewpoint in addition to pose. We find that models trained to jointly to predict viewpoint and pose systematically show significantly improved cross-dataset generalization.

The chapter is based on the work originally published in: **Zhe Wang**, Daeyun Shin, and Charless Fowlkes "Predicting Camera Viewpoint Improves Cross-dataset Generalization for 3D Human Pose Estimation." ECCVW 2020 [182].

**Chapter 4**: We explore the hypothesis that strong prior information about scene geometry can be used to improve pose estimation accuracy. To tackle this question empirically, we have

assembled a novel *Geometric Pose Affordance* dataset, consisting of multi-view imagery of people interacting with a variety of rich 3D environments. We utilized a commercial motion capture system to collect gold-standard estimates of pose and construct accurate geometric 3D models of the scene geometry. To inject prior knowledge of scene constraints into existing frameworks for pose estimation from images, we introduce a view-based representation of scene geometry, a *multi-layer depth map*, which employs multi-hit ray tracing to concisely encode multiple surface entry and exit points along each camera view ray direction. We propose two different mechanisms for integrating multi-layer depth information into pose estimation: input as encoded ray features used in lifting 2D pose to full 3D, and secondly as a differentiable loss that encourages learned models to favor geometrically consistent pose estimates. We show experimentally that these techniques can improve the accuracy of 3D pose estimates, particularly in the presence of occlusion and complex scene geometry.

The chapter is based on the work originally in: **Zhe Wang**, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes "Geometric Pose Affordance: 3D Human Pose with Scene Constraints ." Arxiv 1905.07718 2019 [180].

**Chapter 5**: To better estimate and represent the full 3D shape of the human body, we develop a framework with three consecutive modules. A dense map prediction module explicitly establishes the dense correspondence between the image evidence and each part of the body model. The inverse kinematics module refines the key point prediction and generates a posed template mesh. Finally, an inpainting module relies on the corresponding feature, prediction and the posed template, and completes the predictions of occluded body shape. Our framework leverages the best of non-parametric and model-based methods and is also robust to partial occlusion. Experiments demonstrate that our framework outperforms existing 3D human estimation methods on multiple public benchmarks.

**Chapter 6**: Concludes this dissertation and presents several open directions for future research.

# Chapter 2

# Related Work

## 2.1 Introduction

3D human pose estimation is attracting increasing attention from industry due to its strong application potential in entertainment such motion retargeting [172, 197], animation, hollywood motion capture (3D Avatart), gaming (Netease, Blizzard, SandBox), sport analysis(Second spectrum and Traceup); and also in health care [97] as Autism, Parkinson, physical rehabilitation, pressure generated matte [22] and emotion (Psychology); Beyond those applications, 3D human pose estimation is also strongly connected to other computer vision and robotics topic such as robot learning [73], action anticipation, motion prediction, affordance learning, self-driving cars (motion prediction, trajectories prediction), activity recognition and explanation [125, 93], person generation, priors for segmentation [28], HCI(assist leaving), virtual reality (Holelens2 and Tiotech 7D), Amazon Go, augmented reality, education [12], scene understanding [178, 31], and proxemics recognition [44].

3D pose estimation has also drawn a large amount attention in academia as shown in Fig 2.1, due to its greater ambiguity when compared with 2d pose estimation. This ambiguity

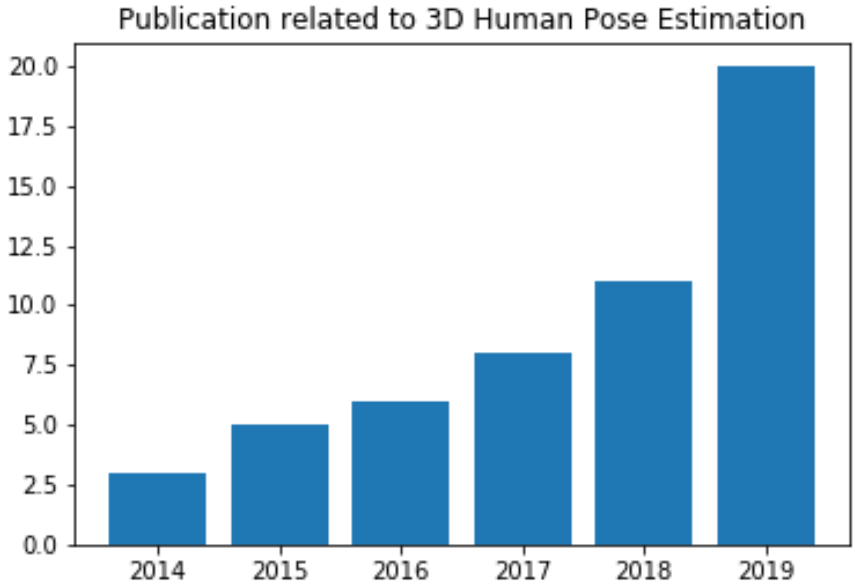
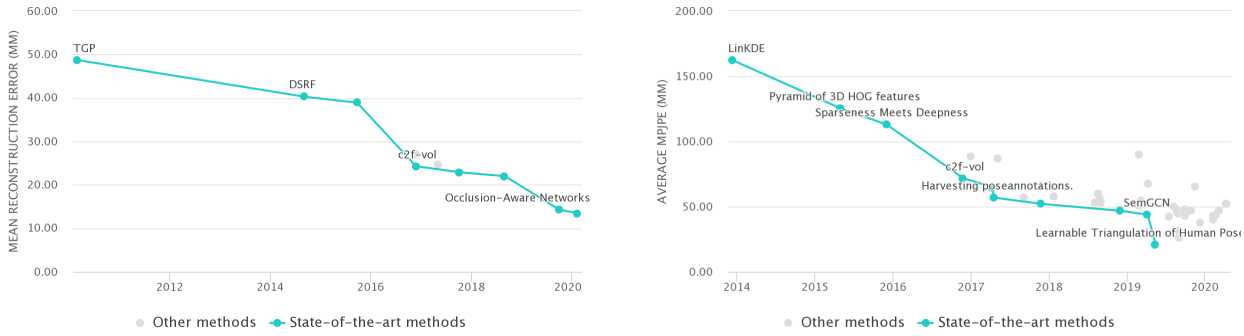



Figure 2.1: Number of 3D human pose paper every year from CVPR/ECCV/ICCV.

makes it easier to incorporate different priors such as geometrical model [113], kinematic model [215] and body shape prior [92]. 3D human pose researchers have designed and constructed datasets using different software and hardware (VICON, The Capture, IMU) as well as different cameras (Kinect, commercial synchronized cameras). These datasets are also captured in different environments (e.g. controlled lab environment and in the wild environment). These datasets vary with respect to body sizes, camera intrinsic and extrinsic parameters and body and background appearance. Deep learning experts have also designed different architectures (1D convolutional neural network [121], graph convolutional networks [213], fully-connected neural networks [98], recurrent neural networks [50] ) and representations (point [98], heatmap [215] and voxel [118]) to not only improve 3D human pose estimation performance, but also accelerate the inference time and reduce model size. The current chapter is motivated by these rapid developments in the last several years.




(a) HumanEva performance vs. years

(b) Human36M performance vs. years

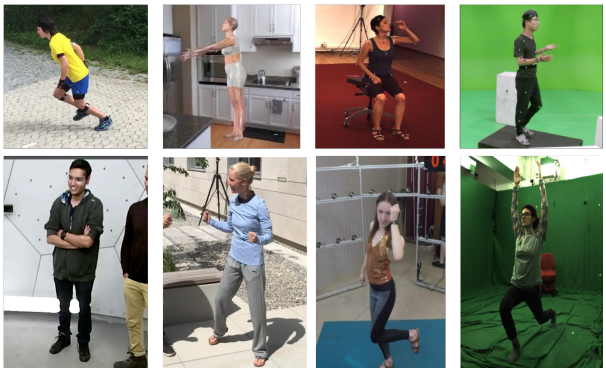
Figure 2.2: We plot performance vs. year from Paper-with-code, on HumanEva [145], and the 45x larger Human36M [52] dataset. We can see even though the performance has saturated on both datasets, the monocular based methods still have about 20mm gap, showing the complexity of Human36M datasets.

### 2.1.1 Scope of this chapter

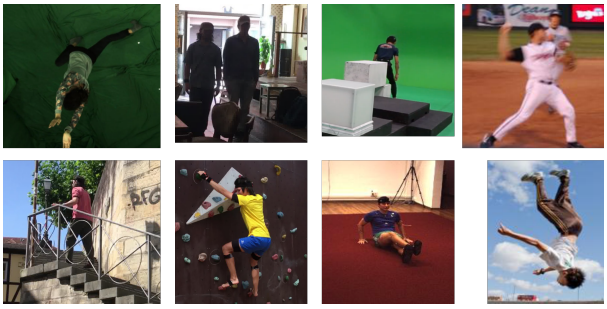
This chapter focuses on major progress made in the last five years, and we restrict our attention to monocular images, leaving the important subject of video pose / multi-view pose as a topic for separate consideration in the future.

The main goal of this chapter is to offer a comprehensive survey of deep learning based 3D human pose estimation techniques, and to present some degree of taxonomy, a high level perspective and organization, primarily on the basis of popular datasets, representations, evaluation metrics, priors, and problems not fully handled. The intention is to make our categorization helpful for readers in obtaining an accessible understanding of similarities and differences between a wide variety of strategies.

The remainder of this chapter is organized as follows. Popular datasets, datasets bias and evaluation criteria are summarized in Section 2.2. We describe how researchers represent 3D human pose in network in Section 2.3. We list and discuss details of the priors that can be used to solve the ill-posed 3D human pose estimation problems in Section 2.4. Widely used one-stage and two-stage architectures are discussed and compared in Section 2.5, and useful



(a) Normal Pose



(b) Extreme Pose

Figure 2.3: 3D human pose estimation algorithm not only needs to handle normal pose cases, but also tackle extreme scene such as rare view point, low lighting, strong scene occlusion, motion blurry, person far from the camera, in the wild images, strong self-occlusion and rare human pose.

codebase is also linked in 2.6. Finally, in the last section, we conclude the chapter with an overall discussion of 3D human pose estimation, and state-of-the-art performance.

| Dataset | Frames | Year | Metric | Feature |
|---|---|---|---|---|
| HumanEva [145] | 80k | 2010 | MPJPE | 1st synchronized 3D pose and image dataset |
| Human36M [52] | 3.6M | 2014 | MPJPE | most polular one, with MR test set |
| MPI-INF-3DHP [101] | 1.3M | 2017 | MPJPE, PCK3D | indoor and outdoor, markerless |
| Total Capture [164] | 1.9M | 2017 | MPJPE | with IMU information and 2d matte |
| SURREAL [171] | 6M | 2017 | MPJPE, PVE | SMPL model, depth, body parts segmentation |
| UnitePeople [77] | 52k | 2017 | MPJPE, PVE | dataset rich annotated with SMPL model |
| 3DPW [173] | 51k | 2018 | MPJPE, PVE | IMUs and phone captured videos, in the wild |
| JTA [27] | 460k | 2018 | MPJPE | Game rendering in ubran scenes, multiple persons |
| GPA [180] | 0.7M | 2019 | MPJPE | affordance learning and full scene geometry |
| PROX [44] | 180 | 2019 | MPJPE | geometry on point cloud, SMPLify-X model |
| GTA-1M [9] | 3M | 2020 | MPJPE | large scale pose and scene context |
| HUMBI [202] | 26M | 2020 | MPJPE | gaze, garment, more subjects, and rich camera views |

Table 2.1: Comparison of existing popular datasets for training and evaluating 3D human pose estimation. Larger datasets with more diverse features are proposed recently to facilitate the development of 3D human pose estimation.

## 2.2 Datasets

Deep learning is data hungry; Therefore numerous amounts of images with humans in various clothes, scenes, lighting conditions, view points, motion blur as in Fig 2.3a and different poses as in Fig 2.3b are required to train a good 3D human pose estimator. Many 3D human pose

datasets with diverse features were proposed after the year 2014 as described in Table 2.1. In this section, we will discuss the differences, techniques and motivation behind the datasets and compare them in details.

### 2.2.1 Getting groundtruth for datasets

**Marker-based motion capture for ground-truth 3D pose** The work of [145] offers the first large-scale 3D human pose estimation dataset with synchronized images and ground-truth 3D keypoint locations. It was captured and solved using the commercial motion capture software called VICON Blade. The VICON system is with cameras covering the capture space. It is used to track the 3D coordinates of IR-reflective markers attached to the surface of subjects and objects. The tracking maintains the label identity and propagates it through time from an initial pose which is labeled either manually or automatically. A fitting process uses the position and identity of each body label, as well as proprietary human motion models, to infer accurate pose parameters. H36M [52] scales their dataset to 3.6 million images covering a wider range of professional subjects and carefully enriches number of actions. They also introduce 4 synchronized commercial high-resolution cameras enabling multi-view study on human pose estimation. To alleviate the heavy dependency on mocap systems, TotalCapture [164] is proposed with multiple viewpoint videos and IMU (inertial measurement unit). This additional IMU sensor enables further study of multi-modal capture of 3D human pose ground truth. A novel geometric pose affordance dataset (GPA) [180] is assembled to explore the hypothesis that strong prior information about scene geometry can be used to improve pose estimation accuracy. The dataset not only provides the 3D pose ground truth, but also curates full scene geometry based on the mocap system. Similarly, PROX [44] not only provides the 3D human joints ground truth but also scene geometry, offering a promising test-bed for the research in 3D human pose estimation with geometric affordance. The geometry provided by PROX is captured by Kinect point cloud while GPA

dataset lets annotators create Maya mesh models which are aligned with the captured marker location.

**Marker-less motion capture for ground-truth 3D pose** To overcome the limitations of marker-based data collection, marker-less approaches are also used. 3DHP [101] relies on the commercial marker-less motion capture software called 'the captury'. As they do not have to rely on special suits and markers, they can capture motions wearing everyday apparel, including loose clothing. They record in green screen studio to allow automatic segmentation and augmentation. In addition, 3DHP covers a wide range of viewpoints including normal camera viewpoints and extreme camera viewpoints as shown in Fig 2.3a and 2.3b, they also capture outdoor images for evaluation. [217] explores motion capture both indoor and outdoor using a Drone. The system only needs an autonomously flying drone with an on-board RGB camera and is usable in various indoor and outdoor environments. Besides the capability of tracking a moving subject, a flying drone also provides fast varying viewpoints, which is beneficial for motion reconstruction. To make motion capture truly unconstrained (both in the wild environment and to avoid moving around to cover full body), [141] uses multiple micro-aerial-vehicles (MAVs), each equipped with a monocular RGB camera, an IMU, and a GPS receiver module. Together with 2d joint detectors, 3D human body model, and a powerful prior on human pose, they successfully demonstrate outdoor full-body, markerless motion capture. However, the number of the views and diversity of race, skeleton size are still limited. HUMBI [202] presents a large multiview dataset to facilitate modeling view specific appearance and geometry of gaze, face, hand, body, and garment from assorted people. 107 synchronized high-definition cameras (70 cameras facing at the front body) are used to capture 772 distinctive subjects across gender, ethnicity, age, and physical condition. 26M images make HUMBI the largest dataset ever.

**Rendering for ground-truth 3D pose** Rendering or game engine is the inverse procedure of 3D reconstruction, which provides alternatives to get free supervision from video games [74] or physical-based rendering. SURREAL [171] generates 6M images together with gruond truth of 3D human shape, 3D body part segmentation, 2d human depth, 2d part segmentation, clothing, camera parameters, human surface normals, optical flow of human motion and even light conditions. This large scale new dataset opens up new possibilities for advancing person analysis using cheap and large-scale synthetic data. However, this dataset lacks human scene interaction (occlusion) and does not have multi-view information. JTA [27] dataset is proposed with a vast number of different body poses, in several urban scenarios at varying illumination conditions, viewpoints, especially occlusion annotation, by exploiting the highly photorealistic video game *Grand Theft Auto V* developed by *Rockstar North*. However, JTA dataset focuses more on urban scene, which is hard explore the affordance learning between scene context and pose prediction. Thus, GTA [9] dataset is curated to predict person future poses and locations, given the scene image and the person's past pose and location history in 2D. The GTA dataset consists of 3M frames and 30k action segments.

**Human-in-the-loop ground-truth 3D pose** It is hard for humans to accurately annotate poses in 3D, but humans can collaborate with pre-defined templates [7, 96] or generative models to roughly annotate and get relatively small error datasets in 3D. In this sense, UnitePeople [77] dataset is proposed based on the collaboration between SMPLify [6] extended with human silhouette, and human annotators. This procedure can generate rich-annotated ground truth labels on in-the-wild images with 3D human joint, 2d part segmentation and even 3D human mesh model. UnitePeople dataset has 52k in-the-wild images. However, they do not cover videos domain. With the introduction of IMU sensors in motion capture [164], 3DPW [173] is able to capture in-the-wild 3D human pose ground truth (extreme lighting, interacting with scene geometry, person far away from cameras). It relies on state-of-the-art 2d joint detector, together with SMPL [92] model and IMU sensors to robustly fit to the image evidence. Even

| Dataset | H36M | GPA | SURREAL | 3DPW | 3DHP |
|---|---|---|---|---|---|
| Imaging Space | 1000 × 1002 | 1920 × 1080 | 320 × 240 | 1920 × 1080 | 2048 × 2048 or 1920 × 1080 |
| Camera Distance | 5.2 ± 0.8 | 5.1 ± 1.2 | 8.0 ± 1.0 | 3.5 ± 0.7 | 3.8 ± 0.8 |
| Camera Height | 1.6 ± 0.05 | 1.0 ± 0.3 | 0.9 ± 0.1 | 0.6 ± 0.8 | 0.8 ± 0.4 |
| Focal Length | 1146.8 ± 2.0 | 1172.4 ± 121.3 | 600 ± 0 | 1962.2 ± 1.5 | 1497.88 ± 2.8 |
| Bone Length | 3.9 ± 0.1 | 3.7 ± 0.2 | 3.7 ± 0.2 | 3.7 ± 0.1 | 3.7 ± 0.1 |

Table 2.2: Comparison of existing datasets commonly used for training and evaluating 3D human pose estimation methods. We calculate the mean and std of camera distance, camera height, focal length, bone length from training set. Focal length is in mm while the others are in unit meters. 3DHP has two kinds of cameras.

though the ground truth has small errors, researchers can still evaluate their method on this dataset.

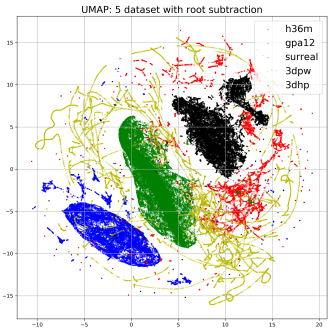
### 2.2.2 Bias for each dataset

We select 5 representative datasets ranging from marker-based capture (H36M [52]), marker-less capture (3DHP [101]), render-based (SURREAL [171]), model-based (3DPW [173]), and with geometry features (GPA [180] (pose tends to be different with geometry around)). We list the bias across different datasets in table 2.2 and Fig 2.4.

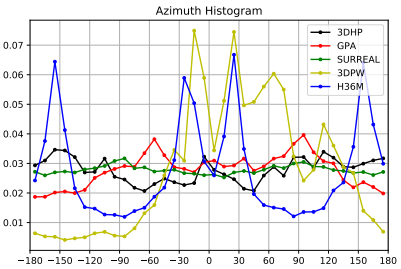
**Imaging space** Different datasets are captured with different cameras, which may result in different size of images and distortion effect. For large images like H36M, GPA, 3DPW , 3DHP, images should be first undistorted before sending to either multi-view algorithm or monocular algorithm, to make the multi-view geometry feasible and neural network avoid overfitting to this distortion effect. Images from SURREAL are in smaller size, thus, the zoom in (SURREAL) and zoom out (3DPW, 3DHP, GPA, H36M) of original image may affect the image quality sent to the neural network. This imaging size bias may harm cross-data generalization. For two-stage algorithms, the different xy magnitude without normalization will also follow the variation of imaging space varies, thus, harm the generalization.

**Camera space** Cameras vary across different datasets, in focal length, camera center, camera height, camera-person distance and view direction. Focal length and center is useful when you back-project the 2d space and relative z prediction to absolute z prediction for calculating MPJPE [107]. This differences lie in camera intrinsic parameter motivates us to handle the back-projection problem without the pre-known camera center and focal length. Camera-person distance determines how far the person is from the camera, and affects the image quality if we want the person to show the same size in both training and evaluation. Camera height correlates with view direction. We use the left shoulder, right shoulder and pelvis to form the body-centered coordinate and treat the camera view point relative to this body-centered coordinate frames. We factorize the view direction into azimuth histogram as in Fig 2.4b and elevation histogram as in Fig 2.4c for 50k sample poses from each of the 5 datasets. We observe **H36M** has a wide range of view direction over azimuth with four distinct peaks ($-30$ degree, 30 degree, $-160$ degree, 160 degree), it shows that during the capture session subjects are always facing towards or facing away the control center while the four RGB cameras captured from four corners. H36M has a clear bias towards elevation above 0; **GPA** is more spread over azimuth compared with H36M, most of the views range from $-60$ degree to 90 degree; **SURREAL** synthetically sampled camera positions with a uniform distribution over azimuth, and also have a uniform distribution over elevation. The viewpoint bias for **3DPW** arises naturally from filming people in-the-wild from a handheld or tripod mounted camera roughly the same height as the subject. Of the non-synthetic datasets, **3DHP** is the most uniform spread over azimuth and includes a wider range of positive elevations, a result of utilizing cameras mounted at multiple heights including the ceiling.

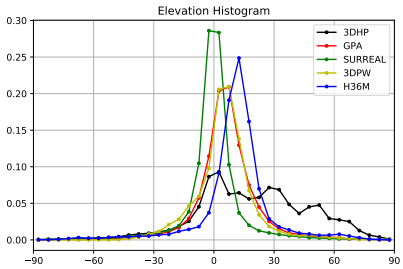
**Pose space** A standard approach is to treat 3D human pose estimation as regressing the 3D joint location relative to the root joint. We list the root joint, especially Z distance (camera distance) in table 2.5. To characterize the variability in pose after the root-joint is




(a) UMAP WITH ONLY ROOT-SUBTRACTION




(b) VIEWPOINT AZIMUTH

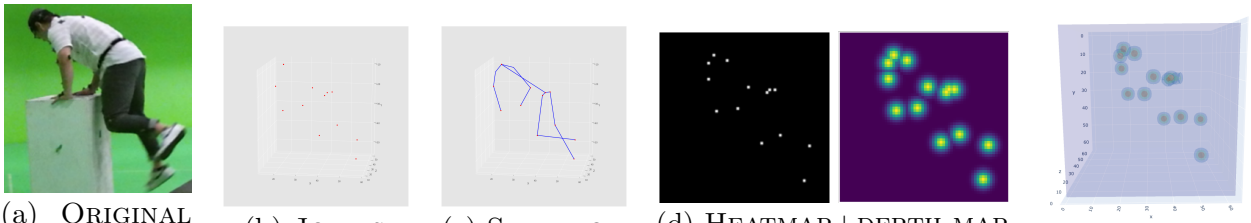



(c) VIEWPOINT ELEVATION

Figure 2.4: (a). Distribution of view-dependent, view-independent body-centered pose, visualized as a 2D embedding produced with UMAP [100]. (b-c). Distribution of camera viewpoints relative to the human subject. We show the distribution of camera azimuth $(-180°, 180°)$ and elevation $(-90°, 90°)$ for 50k poses sampled from each representative dataset (**H36M**, **GPA**, **SURREAL**, **3DPW**, **3DHP**).

factored out, we used the coordinates of 14 joints common to all datasets expressed in the root-relative coordinate frame. To visualize the resulting high-dimensional data distribution, we utilize UMAP [100] to perform a non-linear embedding into 2D. Figure 2.4a shows the resulting distributions which show a the posing difference across the datasets. We further illustrates the skeleton size in table 2.2, which is another prior useful for back-projecting from 2d prediction to 3D space.

**Appearance bias** We list the normal case and extreme case in Fig 2.3. These images appearance differs because of view direction, clothing, lighting and background modeling. The gender, ethnicity, and clothing in HUMBI [202] shows great diversity as HUMBI is captured with numerous amount of subjects. However, in terms of lighting and natural background scenes, images from 3DPW [173] show great variety. Images from JTA [27], GPA [180], PROX [44] and GTA-1M [9] introduce lots of scene occlusion, self-occlusion and scene affordance. Datasets captured in controlled environments (GPA, H36M, 3DHP, TotalCapture, Panoptic Studio) [180, 52, 101, 164, 58] tend to have clean background, making it necessary to train together with 2D datasets like COCO [91] and MPII [2] to make algorithm generalize



(a) Original Image (b) Joints (c) Skeleton (d) Heatmap+depth map (e) Voxel Map

Figure 2.5: The example image from GPA [180] with corresponding common representation for 3D human pose: vector or coordinate (b), skeleton representation (c), 2d heatmap (d) + depth map (e), and voxel map (f).

to in-the-wild images from 3DPW, UnitePeople [173, 77]. Sports images in UnitePeople [77] may have motion blur while rendering image from SURREAL [171] may not be realistic enough and lack of content showing the person interacting with scene context.

## 2.3 Representations

Researchers has proposed different representation as regression target to address 3D human pose estimation problem. The optimization goal of these representation lies in three perspectives: (i), to leverage pictorial structure. (ii), to reduce the memory consumption. (iii), to use the extra 2d pose data. Along this line are point/vector representations in Fig 2.5b, skeleton representation in Fig 2.5c, heatmap + depth map representation in Fig2.5d, and voxel map in Fig 2.5e.

### 2.3.1 Point/Vector/Matrix

**Notation** Given an image $I$ with a human subject in the center as shown in Fig 2.5a, we aim to estimate the 3D human pose represented by a set of 3D joint coordinates of the human skeleton as shown in Fig 2.5b, $P \in \mathbb{R}^{J\times 3}$ where $J$ is the number of joints. We follow the convention of representing each 3D coordinate in the local camera coordinate system

associated with $I$. The first two coordinates are given by image pixel coordinates and the third coordinate is the joint depth in metric coordinates (e.g., millimeters) relative to the depth of a specified root joint. We use $P_{XY}$ and $P_Z$ respectively as short-hand notations for the components of $P$.

**One stage methods** One stage methods directly regress the $P \in \mathbb{R}^{J\times 3}$ given the image $I$ mentioned above. [83] regress the relative distance of one joint relative to its parent $J_i - J_{P(i)}$, and another branch to detect the 2D joint location. Both detection and regression branches are based on fully connected layers, which do not leverage the image structure. [215] initially implement their network to detect $P_{XY}$ using heatmap while regressing $P_Z$ as a normalized vector according to bone length. PoseNet [107, 190] later on leverages integral techniques to convert voxel map differentiably to regress $P_X, P_Y$ separately in image space and $P_Z$ normalized by pre-defined max person range value (1000 mm) and camera intrinsic parameter (focal length and camera center).

**Two stage methods** Two stage methods first extract 2d joint location from images, and then regress these $P \in \mathbb{R}^{J\times 3}$. Different networks use different normalization methods to process the 2d input and 3D output. Simple-baseline [98] treats the task which lift 2d image location to 3D camera coordinates as a machine learning problem. Both input and target are pre-processed with mean subtraction and standard deviation. They also apply advanced residual connection to enhance the simple network. [13] build a 3D pose library to match the detected 2d pose to the nearest 3D pose. [109, 126] represent both the 2D and 3D human poses using $N \times N$ distance matrices, and formulate the problem as a 2D-to-3D distance matrix regression. By enforcing positivity and symmetry of the predicted matrices, the approach also has the advantage of naturally handling missing observations and allow to hypothesize the position of non-observed joints. Graph neural network [213] builds the connection between joints and leverages the state-of-the-art graph convolutional network to

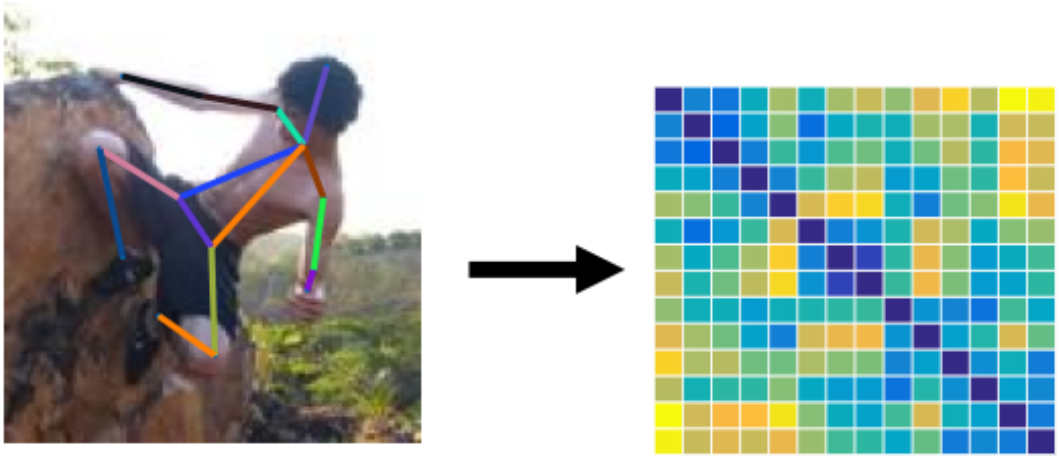


Figure 2.6: The sample image with corresponding 2d distance matrix. (Image credit: [109])

refine them. Videopose [121] leverages 243 frames and 1D dilated convolution architecture to model the poses in temporal domain. Similar temporal modeling based on pose vector representation is also seen in LSTM [79] and Fully-connected neural network [50, 23]. We will compare these architectures in detail in Section 2.5.

**Distribution** Lifting from 2d to 3D is an ill-posed problem because of depth ambiguity and occluded joints. [80] propose a novel approach to generate multiple feasible hypotheses of the 3D pose from 2D joints. By modeling the 3D pose space with guassian mixture model (mixure density), they predict prior, mean, and covariance of the pre-cluster 3D pose mixture. [148] employs the multimodal distributions prediction idea in a one-stage method and train on image with 2d joint label and 3D joint label together.

**Root joint** Most 3D pose estimation methods always treat root joint (pelvis) as known location. However, this is not true in real scenarios. RootNet [107] uses ResNet [46] with deconvolution and pin-hole camera model to localize the root joint ($X, Y, Z$ location of pelvis in camera coordinate). Videopose [121] uses both 2D data and 3D data to predict root location in a semi-supervised way. Concurrent work [5] also estimates the root joint and

resolves the ambiguities/uncertainties in outdoor KITTI dataset using monte carlo dropout and Laplace distribution priors.

### 2.3.2 Heatmaps

Following the 2d pose estimation work trend, 3D pose regression also moves from joint regression using fully-connected layer [163] to joint heatmap [161] regression. Fig 2.5d shows a target distribution created from ground-truth $P$ by placing a Gaussian with $\sigma = 3$ centered at each joint location. At inference stage, the 2d joint heatmaps are decoded to x,y joint locations using an argmax function. It is either used together with Starmap (canonical view heatmap as shown in Fig 2.5d) [216] in the second implementation in [215], or with a location map as in VNect [104]. Both Starmap and location map are proposed to have the 3D pose prediction linked more strongly to the 2D appearance in the image. The $P_X, P_Y, P_Z$ values are read off from their respective location-maps at the position of the maximum of the corresponding joint's 2D heatmap. [160] takes an integrated approach that fuses probabilistic knowledge of 3D human pose with a multi-stage CNN architecture and uses the knowledge of plausible 3D landmark locations to refine the search for better 2D locations.

### 2.3.3 Voxels

Even though heatmap and depth map is related in space (xy aligned), their correlation in z space is not carefully exploited. Then, how about defining a 3D spherical gaussian voxel as regression target (as shown in Fig 2.5e)? [118] is the first work applying voxel in deep 3D human pose estimation, they propose a fine discretization of the 3D space around the subject and train a ConvNet to predict per voxel likelihoods for each joint. They also employ coarse-to-fine prediction scheme, multi-task learning to leverage 2d pose data, and achieve promising results on in-the-wild images. However, quantizing 3D location into a heatmap has

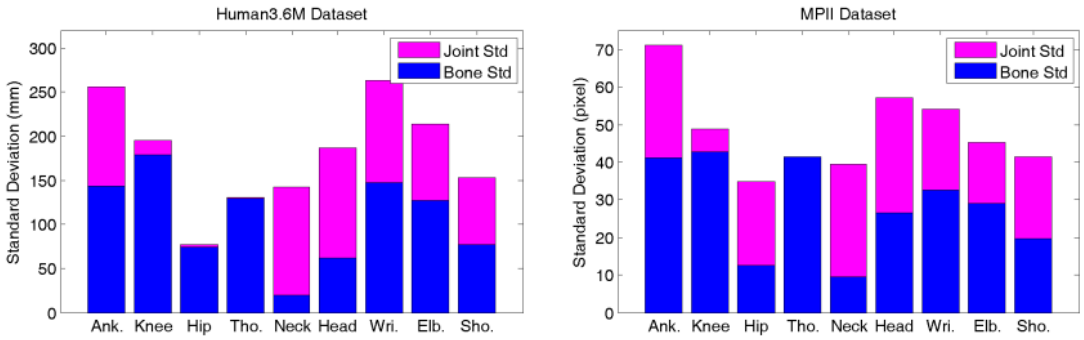



Figure 2.7: Standard deviations of bones and joints for the 3D Human3.6M dataset and 2D MPII dataset. (Image credit: [150])

| | Mean(cm) | Max(cm) | Std(cm) |
|---|---|---|---|
| 6D | 1.9 | 28.7 | 1.2 |
| 5D | 2.0 | 33.3 | 1.4 |
| Quat | 3.3 | 87.1 | 3.1 |
| AxisA | 3.0 | 120.0 | 2.3 |
| Euler | 2.7 | 48.7 | 2.1 |
| Matrix | 22.9 | 53.6 | 4.0 |

Table 2.3: Empirical results for human body inverse kinematics test. It shows that the 6D representation performs the best with the lowest errors and fastest convergence. Table credit [221].

its inherent quantizing error and the voxel target is very memory consuming. [151] solves the quantizing error by introducing soft-argmax to make the voxel to joint process differentiable. [112] marginalizes the xyz voxel to xy, yz, xy heatmap space. They reduce the memory consumption significantly while achieving better performance.

### 2.3.4 Skeleton Representation

"Bones are more stable than joints and easier to learn." – claimed by [150]. They propose a structure-aware (representing target as bone) regression method and demonstrate its effectiveness on both 2d and 3D pose datasets. They reparameterize the joint as the bone following $B_k = J_{parent(k)} - J_k$, in addition, they also enforce long-range geometric constraint by training with both bone ground truth and joint ground truth. Similarly, [218] formulates

a person as a kinematic tree: starting from the root joint, the child joint is represented with rotation matrix and translation. With the developed kinematic layer, they can train together with motion parameters (rotation and translation), joint loss and model fitting loss.

**Rotation representation** [218] regress rotation uses the euler angle, however, rotation angle itself is not numerically stable. [124] discuss several rotation representations and drawbacks: rotation matrices, euler angles, quaternions, axis-angle, for the corresponding kinematic chain. Recently, a new 6D rotation representation [221] is proposed, and demonstrates numerical stablily and continuous both in theory and practical for neural network to learn. In terms of rotation representation selection, [182] and [223] selects quaternions but treat loss differently; [6, 69] uses rotation matrix; [116, 44] picks up axis-angle while [68, 64] votes for 6D representation.

### 2.3.5 Multi Person Association

**Bottom-up approaches** [204] propose a bottom-up method called MubyNet and deep volume encoding to handle body joint detection, person grouping, and pose and shape estimation together by integrating representation based on 3D reasoning at all stages. This avoids suboptimality resulting from separate 2d and 3D reasoning, and uses the combined representation for grouping. [26] compress the multi joint voxel into one voxel, thus making the memory consumption $1/number\_joint$ of original voxels [118] representation. They also extend this representation to a multi-person setting.

**Top-down approaches** Unlike MubyNet, LCRNet [139] is a top down method and proposes joint anchors. LCRNet is an end-to-end architecture for joint 2D and 3D human pose estimation in natural images. The ability of generation and scoring of a number of pose proposals per image, allows LCRNet to predict 2D and 3D poses of multiple people

simultaneously. By treating pose anchors as class centers, it can also handle joints that are partially visible. [107] follow the top down pipeline and breaks it into two problems: root prediction and pose prediction. It can handle the root joint prediction without the need for camera intrinsic parameters and instead bases prediction on the pinhole camera model and mask-rcnn detection area. They also propose a new metric called MRPE. [103] propose multi-person interaction and occlusion dataset: MuCo-3DHP and MuPoTS-3D, and novel occlusion-robust read-out pose-maps which enable full body pose inference even under strong partial occlusions by other people and objects in the scene. [102] divide the multi-person 3D human pose estimation into 3 stages: 2D key point detection, part affinity field [10]; person localization; identity tracking and temporal kinematic fitting. Their network can handle partial occlusion, generalizes to diverse scenes, and runs real time. Their 3D pose encodes local kinematic context, which contains person-person interaction and interaction between parent joints and child joints.

## 2.4 Priors

Estimating 3D human pose from monocular images or videos is an ill-posed problem which can benefit from prior constraints on prediction. In this section, we will talk about constraints from temporal smoothness, human shape, human kinematics, to more modern scene constraints.

### 2.4.1 Temporal Modeling

[172] utilize cycle consistency and velocity smoothness to stabilize the motion retargeting. [86] use sparse annotations and automatically collect the annotations across the entire video by solving the 3D trajectory completion problem. By fine tuning the model, they get decent performance in this semi-supervised setting. [50] explore temporal information by designing

| Methods | Frames | Input | Networks |
|---|---|---|---|
| [90] | 10/5s | Images | CNN |
| [121] | 243/5s | Keypoints | CNN |
| [62] | 10/0.4s | Features | CNN |
| [19] | 128/2.6s | Keypoints | CNN |
| [23] | 20/0.4s | Keypoints | NN |
| [50] | 5/0.1s | Keypoints | RNN |
| [79] | 3/0.3s | Keypoints | RNN |

Table 2.4: Temporal length, input and neural network type to model 3D human pose. For [23, 50] we did not find whether they downsample the videos or not, so we assume they use the 50hz H36M videos for training.

a sequence-to-sequence network composed of layer-normalized LSTM units with shortcut connections connecting the input to the output on the decoder side and impose a temporal smoothness constraint during training. They also show their model better than traditional smoothing algorithms like median filter and that their model is robust to gaussian noise. [79] propose a new long short-term memory (LSTM)-based deep learning architecture, where each LSTM is connected sequentially to reconstruct 3D depth from the centroid to edge joints through learning the intrinsic joint dependency. [23] present a simple temporal network that exploits temporal and structural cues present in predicted pose sequences to temporally harmonize the pose estimations. [90] exploit rich spatial and temporal long-range dependencies among body joints for accurate 3D pose sequence prediction and presents a Recurrent 3D Pose Sequence Machine (RPSM) to automatically learn the image-dependent structural constraint and sequence-dependent temporal context by using a multi-stage sequential refinement.

Many kinds of networks are able to incorporate temporal information to the learning: CNN [121], RNN [79] and NN [23]. However, the difference of the length of temporal dependency depends on the input and network type. We list how many frames each method can cover in Table 2.4. From the table, we can see images input [90] can model more sparse frames compared to keypoints input [121]. 1D CNN always models longer than RNN.

| Methods | Paired sup. (MV: multi-view) | | | Unpaired 2D/3D pose Supervision | Sup. for latent to 3D pose mapping |
|---|---|---|---|---|---|
| | MV pair | Cam. extrin. | 2D pose | | |
| [133] | ✔ | ✔ | ✗ | ✗ | ✔ |
| [66] | ✔ | ✗ | ✔ | ✗ | ✗ |
| [16] | ✔ | ✗ | ✔ | ✗ | ✔ |
| [174] | ✗ | ✗ | ✔ | ✔ | ✗ |
| [14] | ✗ | ✗ | ✔ | ✔ | ✗ |
| [75] | ✗ | ✗ | ✗ | ✔ | ✗ |

Table 2.5: Characteristic comparison of weakly-supervised human 3D pose estimation works, in terms of access to direct (paired) or indirect (unpaired) supervision levels. (Table credit: [75])

### 2.4.2 Multi-view Constraint

Although this chapter focuses on monocular 3D human pose estimation, there are numerous research collecting supervision from others views, self-supervised learning or weakly-supervised / unsupervised learning. This research is worth discussing as these setting launch the connection between geometry and learning. [198] propose a new differentiable representation of the epipolar constraint called epipolar divergence – a generalized distance from the epipolar lines to the corresponding keypoint distribution. Epipolar divergence defines how big the error is when it is projected on the other view. [134] propose a method to estimate camera pose jointly with human pose, which enables utilizing multi-view footage where calibration is difficult, by utilizing the view consistency from multi-view cameras they make it effective in predicting 3D human pose. [133] use known camera transformation matrix and implicitly disentangle the foreground and background using unsupervised learning, and this representation is easily transferred to 3D human pose task. [132] comprise three layers of abstraction to represent human subjects: spatial layout in terms of bounding-boxes and relative depth; a 2D shape representation in terms of an instance segmentation mask; and subject-specific appearance and 3D pose information. By collecting supervision from multi-view data, it works for multiple persons and full-frame images, and can then be effectively

leveraged to train a 3D pose estimation network from small amounts of annotated data. Instead of segmenting foreground and background in image space, [16] rely on 2d human pose for self-supervised learning with the same flavor. [66] use off-the-self 2d pose detectors to detect 2d pose in each view and use epipolar geometry to collect supervision from two views. [75] leverage the prior knowledge on human skeleton and poses in the form of a single part based 2D puppet model, human pose articulation constraints, and a set of unpaired 3D poses. Their differentiable formalization, bridging the representation gap between the 3D pose and spatial part maps, not only facilitates discovery of interpretable pose disentanglement, but also allows us to operate on videos with diverse camera movements. We also list the table from [75] as in Table 2.5 illustrating several weakly-supervised approaches utilizing varied set of auxiliary supervision other than the direct 3D pose supervision. There are also several works directly working on fusing information from multiple views [127, 165, 119, 54, 191, 212, 130, 94], however, not in the scope of this chapter.

### 2.4.3 Human Structure Prior

**Human shape prior** SMPL (a skinned multi-person linear model) [92] model is proposed with population of captured human mesh (CAESAR dataset). SMPL is a skinned vertex-based model that accurately represents a wide variety of body shapes in natural human poses. The parameters of the model $M(\beta, \alpha, \gamma)$ are learned from data including the rest pose template, blend weights, pose-dependent blend shapes, identity-dependent blend shapes, and a regressor from vertices to joint locations. The body model is parameterized by shape $\beta$, pose $\alpha$, and translation $\gamma$. The output of the function is a triangulated surface $M$ with 6,890 vertices. [6] first propose an optimized-based approach to directly reconstruct 3D human mesh from a single image with 2d key point detection. [77] fit the human mesh model [92] with in the wild images, together with sihoutte and human-in-the-loop data cleaning, and presents UP3D dataset. [114] integrate SMPL within a CNN, leveraging reliable bottom-up semantic

body part segmentation and robust top-down body model constraints. [51] fit SMPL to multi-view videos. [61, 120] introduce deep learning based end-to-end training models, which can directly predict the SMPL parameters from monocular images. A couple of methods based on HMR further exploit the temporal context to help build more precision and smoothness in a human mesh model. [3] harvest internet videos, and train their model on unlabeled video with pseudo-ground truth 2D pose obtained from an off-the-shelf 2D pose detector. By modeling the feature evolution with temporal encoder and hallucinated motion, they get smoother mesh predictor. [152] build the self-attention based temporal convolution network to efficiently exploit the short and long-term temporal cue. [68] combine optimized-base and learning-based methods by constructing the model-fitting loop. [64] leverage large-scale human mesh datasets (AMASS [95]) to serve as a motion regularizer instead of directly modelling dynamic tissues. Recently, [116] build SMPL-X model to holistically model face, body and hands.

**Kinematics** Human kinematics includes two problems: forward kinematics (FK) and inverse kinematics (IK). **FK** regards the human skeleton as a kinematic tree composed of fixed-length bone and rotation between bones. [1] explore joint angle constraints in 3D to penalize unnatural rotation. [218] model the kinematics with introduced motion parameter and kinematic layers. [215] improve the performance by adding constant bone ratio constraint and generalizing 3D human pose estimation to in-the-wild images. [196] propose an anthropometrically adversarial network as a regularizer. [29] model these kinematics, symmetry between left/right human part and motor control skeletons using an RNN when predicting 3D human joints directly from 2D key point. [150] supervise the training of the network with another bone representation. This representation share the same flavor as part affinity field [10], however, in the same person. [23] proposes two anatomically inspired loss functions to penalize illegal human poses. **IK** is the inverse process of FK: given the set-up pose and skeleton, FK solves the rotation between these skeletons. [172] leverage adversarial

priors to correct unreleastic animation. [174] avoid the overfiting for lifting from 2D to 3D by ignoring 2D to 3D correspondences. They learn a mapping from a distribution of 2D poses to a distribution of 3D poses using an adversarial training approach. By additionally consider estimating cameras, they generalize well to unknown data. [168] propose adversarial inverse graphics networks (AIGNs): weakly supervised neural network models that combine feedback from rendering their predictions, with distribution matching between their predictions and a collection of ground-truth factors. They apply AIGNs to 3D human pose estimation and 3D structure and egomotion estimation, and outperform models supervised by only paired annotations.

### 2.4.4 Pose Templates

[13] build a large 3D human pose set, and treat lifting from 2d to 3D as a matching problem. [137] utilize cmu-mocap data, part-based pixels, and mosaic to composite synthetic datasets and train a robust 3D human pose estimator. [138, 139] use the datasets generated in [137] and cluster the 2d poses/ 3D poses using kmeans. They treat the pose clusters the same as object detection 'anchors', by proposing, classifying, and regressing, they get a robust multi-person 3D human pose estimator. To generate scene afforded poses, [178, 85] treat each pose cluster center as a hidden state in the variational auto encoder. Pose template ideas also apply for egocentric pose estimation when majority part of human is out of view [55]. [158] introduce a Deep Learning regression architecture for structured prediction of 3D human pose from monocular images that relies on an overcomplete auto-encoder to learn a high-dimensional latent pose representation and account for joint dependencies.

### 2.4.5 Ordinal Constraints

Ordinal constraints constrain pose prediction with some reference. It can be a physical space like some key point should be within range of -1000 mm to 1000 mm, or an embedding constraint. [117] uses ordinal depth between joints as reference. They annotate each image and all joints pairs with relative depth. By integrating this information and relying on additional MPII and LSP datasets, they achieve better performance. [214] relax this relative depth by only modeling pairs with local triplet heatmaps. Their HEMLet representation leverage both image structure and weak depth information. [84] train image-pose embedding and score function together with a maximum-margin cost function, the positive pairs will have a higher score compared to negative pairs, which share similar flavor of ordinal constraint in embedding space, which is also shown in [84].

### 2.4.6 Viewpoint Constraints

Number of views in each mocap dataset is limited as shown in Fig 2.4b, 2.4c. [113] utilize traditional structure from motion and explicitly factors viewpoint changes to improve 3D human pose estimation performance in self-supervised setting. [42] embed local regions into a learned viewpoint invariant feature space. Their multi-task framework is able to selectively predict partial poses in the presence of noise and occlusion, however, they work on depth map instead of rgb images. [182] cluster viewpoint from five popular datasets and generate quaternion clusters. By predicting these quaternion cluster and 3D human pose together, they achieve state-of-the-art performance in several datasets and decrease error by a large margin on cross-dataset evaluation setting. There are also recent works [191, 123] formulating viewpoint selection/fusion as a reinforcement learning or meta learning problem.

### 2.4.7 Scene Constraints

This general notion of scene affordance has been explored as a tool for understanding functional and geometric properties of a scene. [43] first reconstruct the static background and the position of each camera using structure-from-motion (SfM). Then they capture 3D human pose in three carefully selected scene: indoor-climbing, dancing in a halfpipe, and running and jumping in an outdoor scene. These pre-captured geometry helps to resolve the ambiguities induced by impossible views from the back side of the climbing wall, partial occlusion and fast movement. [55] exploit cues from the dynamic motion signatures of the surrounding scene and introduces a novel energy minimization scheme to infer the pose sequence to infer the "invisible pose" of a person behind the egocentric camera. [85] build a fully automatic 3D pose synthesizer that fuses semantic knowledge from a large number of 2D poses extracted from TV shows as well as 3D geometric knowledge from voxel representations of indoor scenes. And they introduce a 3D pose generative model to predict semantically plausible and physically feasible human poses within a given scene based on the constructed data. [203] leverage multi-task learning as well as parametric human and scene modelling, to guide semantic representations at both model and image level, and integrate scene constraints including ground plane support and detecting simultaneous volume occupancy by multiple people. [44] capture PROX dataset (180 images) which have point cloud representation of scene geometry and fit the SMPL [92] model with segmented point cloud. By considering inter-penetration and contact constraints, the estimated accuracy of vertex (human mesh) and joint is largely improved. [180] collect Geometric Pose Affordance dataset with 0.7 million images. The dataset has multi-view video and is captured in motion capture studio. Subjects interact with scene geometry in various ways and scene geometry is represented as novel multi-layer depth. [180] is end-to-end trainable and runs much faster than [44].

## 2.5 Architecture

This section provides an overview of some of the most prominent deep learning architectures used by 3D pose estimation community, including single-stage (Hourglass [111], SimpleBaseline [98], and HRNet [149]) as well as two-stage architectures (CNN/RNN/GCN).

### 2.5.1 Single-stage networks

**Single-Stage CNN [83, 163]:** Singe-stage CNNs (before 2016) 2.8 follow how image classification use the network [46], and translate the images into a vector. After that they use the fully connected layer to regress the root-relative coordinate or parent-relative coordinate [84].

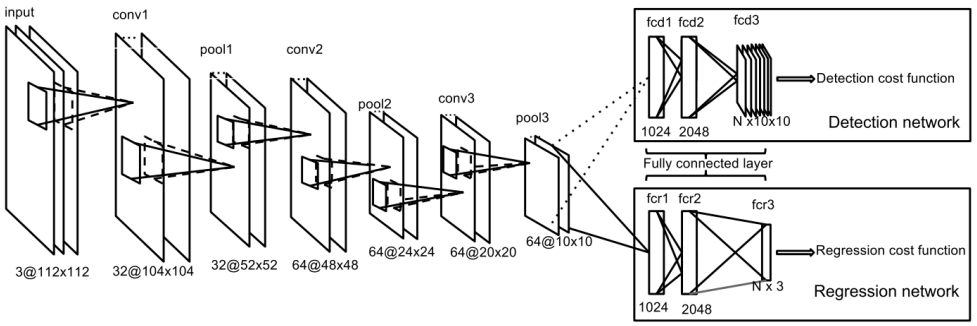



Figure 2.8: Early stage network.

**Hourglass [111]:** The design of the hourglass network ( as shown in Fig 2.9) is motivated by the need to capture information at every scale. While local evidence is essential for identifying features like feet and hands, a final pose estimate requires global context. The person's orientation, the arrangement of their limbs, and the relationships of adjacent joints are among the many cues that are best recognized at different scales in the image. The hourglass captures information at every scale. This way, global and local information are captured completely and are used by the network to learn the predictions. Several 3D human

pose estimation frameworks are based on hourglass backbones. [215] build their network on top of the hourglass network, with two head modules: 2d pose estimation module and depth regression module. They normalize the 2d keypoint ground truth and 3D relative depth consistently to generalize their network to in-the-wild images. [196] treat hourglass network as the backbone to generate 3D human pose while adding a following discriminator for adversarial learning. [118] utilize the strong power capacity of hourglass network to lift 2d heatmaps to 3D voxels in a coarse-to-fine manner.

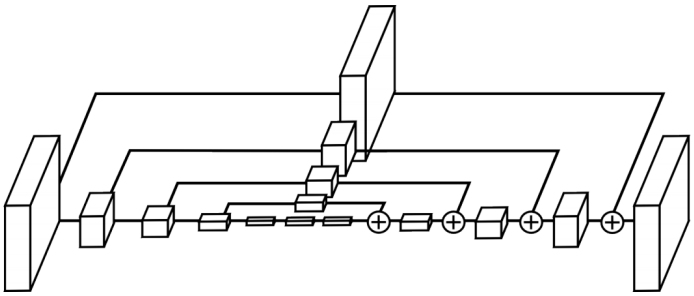


Figure 2.9: Hourglass Network.

**Simple Baseline [190]:** The network structure (as shown in Fig 2.10) is quite simple and consists of a ResNet + few deconvolutional layers at the end. While the hourglass network uses upsampling to increase the feature map resolution and puts convolutional parameters in other blocks, this method uses deconvolutional layers and combines with Resnet [46] backbone in a very simple way. [190] demonstrate its strong power in both 2d pose estimation and pose tracking. [138] originally adopted VGG [146] as the backbone, but later developed a newer version which is based on ResNet [46] backbone, to build a multi-person robust 3D human pose estimator. The code base of [215] is originally based on Hourglass, but they turned to simple baseline later. [104] utilize resnet to extract features and smooth them using kinematics constrain. They also propose spatial-aware location map to regress, making their network robust without referring to voxel representation. [151] is the first attempt combining simple baseline backbone with 3D human pose, and serves as the backbone for the winning solution in ECCV 2018 3D human pose estimation challenges [154]. [107] extend this solution

to estimate both relative 3D human pose and the root position using the pin-hole camera model. [214] hack the feature upsampling part in the simple baseline backbone and propose part-centric heatmap triplets to enforce local depth constraint between parent-children joints.

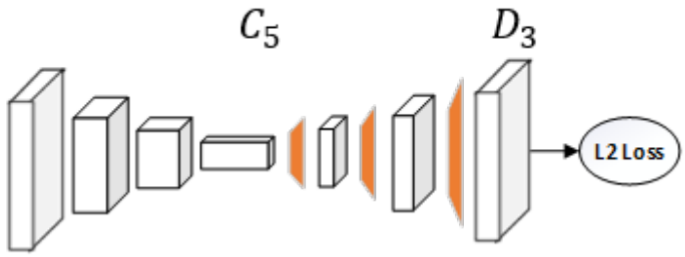



Figure 2.10: Simple Baseline Network.

**HRNet [149]:** Another popular model in this category is the recently developed pose estimation network, high-resolution network (HRNet, as shown in Fig 2.11). Other than recovering high resolution representations as done in Simple-Baseline, HRNet maintains high-resolution representations through the encoding process by connecting the high-to-low resolution convolution streams in parallel, and repeatedly exchanging the information across resolutions. As HRNet is relatively new, there are few 3D human pose estimation works using HRNet as the backbone to exploit contextual information such as self-attention.

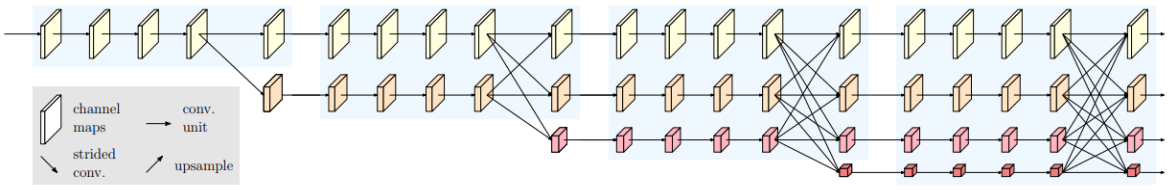



Figure 2.11: Structure of high-resolution network.

**Pre-training Selection:** Pre-training secures a good initialization and helps the networks optimize to a better solution. [190] are pre-trained on ImageNet, which is target for general image classification. [111] are pre-trained on MPII to generalize to articulated human pose

| FLOPs(G) | Network | Pre-training | MPJPE |
|---|---|---|---|
| 9.06 | Hourglass | No | - |
| 9.06 | Hourglass | MPII | - |
| 5.37 | ResNet50 | No | 125.4 |
| 5.37 | ResNet50 | ImageNet | 99.1 |
| 5.37 | ResNet50 | MPII | - |
| 5.37 | ResNet50 | MPII+ImageNet | - |
| 10.24 | HRNet-W32 | MPII+ImageNet | 81.2 |

Table 2.6: The computation complexity (Flops, all input image size as 256 $\times$ 256, only calculate backbones), and how choice of pre-training and backbone selection influence 3D human pose estimation. Training and testing details follow [107]

estimation. 3D human pose estimators are always trained together with MPII [2] datasets to make the network more robust to different human viewpoints, occluded joints, etc. To work on harder datasets such as PoseTrack which includes lots of small person and motion blur, [190] is pre-trained on COCO dataset. It is shown pre-train gives better pose estimation in [149] and the same applies for larger image input size.

We run [107] baseline and list the performance with different backbones and pre-training as in table 2.6. It is both trained and tested on 3DPW dataset, with number of training images 22,375 and test images 35,515. Additionally, we calculate the computation burden (FLOPS) as in the same table. The training and testing follows the original PoseNet [107] paper, which uses ground truth bounding box, with extra MPII dataset to train together. During testing, we have the ground truth intrinsic parameters. During training and testing we make it consistent with input image height and width as 256.

### 2.5.2 Two-stage networks

Two stage architectures build upon the pre-trained 2d detection network which can provide 2d keypoint detection in 2d image coordinates. The second stage networks utilize the 2d single frame keypoint or key points within a video sequence to lift to 3D keypoints in root-relative

| Model | Parameters | FLOPs | MPJPE |
|---|---|---|---|
| [98] | 4.29M | 4.29M | 42.5 (62.9) |
| [213] | 0.43M | 0.43M | 43.8 (60.8) |
| [21] | 1.85M | 1.85M | 57.80 |
| [49] | 16.96M | 33.88M | 41.6 |
| [121] | 16.95M | 33.87M | 37.8 |

Table 2.7: The second stage models, the computation complexity (FLOPS), number of parameters and the corresponding MPJPE in mm.

coordinates.

**Single frame lifting networks** [98] is the first lifting network that lift 2d image keypoints location to 3D root-relative space. They apply the state-of-the art batch normalization, relu, dropout to build basic blocks, and use skip-connection between basic blocks. These networks achieve state-of-the-art performance without seeing any visual information. They also do an interesting ablation study showing how networks perform with noisy or perfect 2d keypoint detection. However, this network treats input 2d pose and output 3D pose as vector without considering the kinematics constraint between them. [29] propose pose grammar which are built by a hierarchy of Bi-directional RNNs (BRNN) to explicitly incorporate a set of knowledge regarding human body configuration (i.e., kinematics, symmetry, motor coordination). By augmenting training samples with virtual views, they achieve state-of-the-art performance. Graph convolutional neural networks [213, 21] are proposed later on to represent each semantic keypoint as nodes, and the kinematics connection as edges. To model the uncertainty in the lifting process, [80] train guassian mixture models using the lifting network.

**Sequence lifting networks** [50] design a sequence-to-sequence network composed of layer-normalized LSTM units with shorcut connections. They also apply temporal smoothness constraint during training for both input and output. The temporal consistency can help

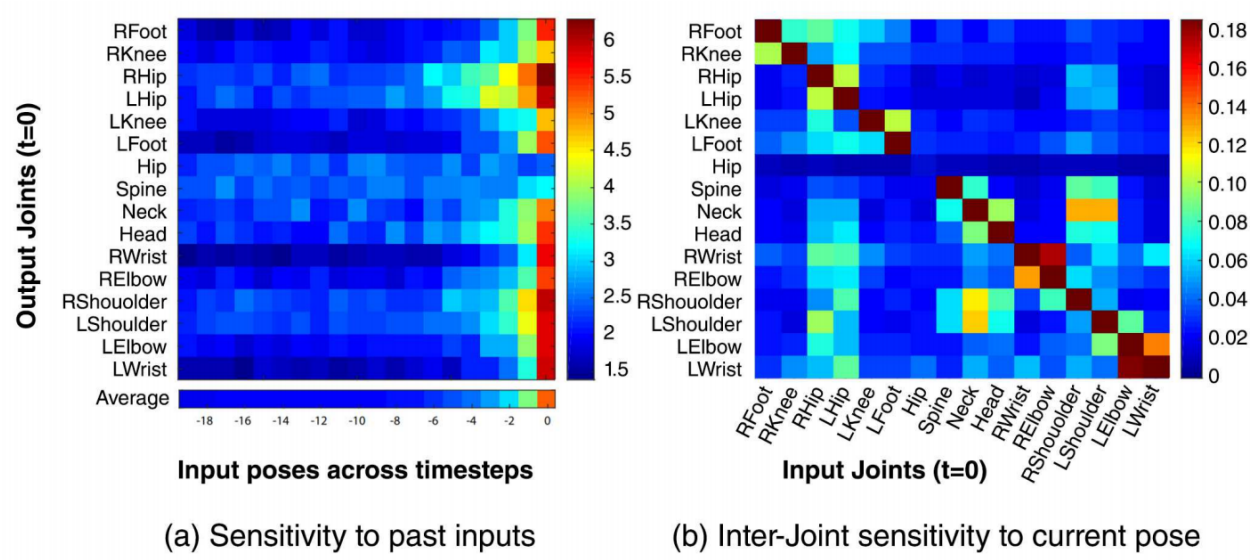



Figure 2.12: Temporal dependency and inter-joint dependency from temporal posenet (TP-Net). Image credit [23]

recover pose even when 2d pose detector fails. TPNet [23] build upon [215] while adding structure loss and builds temporal modeling on the top. TP-net can help diagnosis which frames the current frame relied on, and which joint it relies on most (as shown in Fig 2.12). Based on joint interdependency (JI), [79] propose a new long short-term memory (LSTM) based network called propagating LSTM to reconstruct 3D depth from the centroid to edge joints through learning the intrinsic JI. [121] utilize 1D dilated convolution to model long-term dynamics of the lifting process. This network is semi-supervised use the projection loss and can be extended to estimate root joint location.

We also report the number of parameters, and computational complexity in table 2.7 if code is available.

## 2.6 Benchmarks

In this section, we list the common benchmark and performance comparison on the five representative datasets: Human36M (table 2.8 and 2.9), GPA (table 2.10), SURREAL (table

2.11), 3DPW (table 2.13), and 3DHP (table 2.12). We list the attributes of each method: evaluation metric, backbones, stages, whether they use extra data for training as well as the highlights of each method. We rank the methods with the corresponding evaluation metric. The evaluation metric for H36M, GPA and SURREAL is MPJPE (Mean Per Joint Position Error, in mm unit); 3DHP is evaluated with PCK3D (MPJPE with a threshold of 150mm) with threshold of 150 mm; To cover more methods in the table, 3DPW is evaluated using PA-MPJPE (in mm unit).

**For video-based methods** We can observe the MPJPE decrease from 126.5 in 2016 to 40.1 in 2020 on H36m, the PCK3D increase from 79.4% in 2017 to 93.2% in 2020 on 3DHP, the PA-MPJPE decrease from 92.3 mm in 2018 to 51.9 mm in 2020 on 3DPW, with the introduction of more powerful temporal modeling 1D dilated convolution, more datasets as training (AMASS, Instavariety, Kinetics) and stronger augmentation.

**For image-based methods** MPJPE on H36M decreases from 121.3 in 2015 to 39.9 in 2019, due to the more powerful backbones and stronger constraint and intermediate representation. For GPA dataset, viewpoint matters to improve the performance to state-of-the-art, which is the same for SURREAL datasets. For 3DHP datasets, PCK3D increases from 72.9% in 2017 to 93.2% in 2020 due to the usage of the state-of-the-art backbones and more image-aligned supervision (limb depth map). However, for 3DPW datasets, the error reduction is more attributed to more datasets and more powerful models (SMPL).

## 2.7 Conclusions

In this chapter we discuss how 3D human pose experts curate datasets from different aspects. In addition, we also discuss how deep learning experts design networks, propose

representations, incorporate genetic priors to handle 3D human pose estimation in deep learning era. We have a web link that is updated regularly with awesome 3D human pose papers: [1]

[1] Awesome-3D-human-pose: https://github.com/wangzheallen/awesome-human-pose-estimation#3D-pose-estimation

| Method (H36M) | MPJPE | Backbone | Stages | Extra data | Highlights |
|---|---|---|---|---|---|
| **Video Sequence as Input** | | | | | |
| [18] | 40.1 | 1DCNN | Two | N/A | Occlusion Augmentation |
| [19] | 42.9 | 1DCNN | Two | N/A | Occlusion modeling, cylinder model |
| [87] | 46.6 | MLP | Two | N/A | matrix factorization for sequential 3D human poses |
| [199] | 46.7 | MLP,RNN,CNN | Two | N/A | chirality transform |
| [121] | 46.8 | 1DCNN | Two | N/A | first 1D dilated for convolution sequential 3D human poses |
| [8] | 48.8 | GCN | Two | N/A | loca-to-global GCN |
| [23] | 52.1 | Hourglass | One | MPII | generate rotation-valid pose and explore temporal dependence |
| [79] | 52.8 | RNN | Two | N/A | explore temporal/joint dependence |
| [93] | 53.2 | Inception-V4 | One | MPII | multi-task with action/2d pose |
| [152] | 59.1 | Resnet50 | One | MPII,LSP | skeleton-disentangled representation |
| [3] | 63.3 | Resnet50 | One | AICH,Penn Action 3DHP, COCO LSP, MPII, Flickr | In the wild human shape reconstruction |
| [102] | 63.6 | CNN | One | MPII, LSP | memory-efficient representation |
| [64] | 65.6 | RNN | Two | 3DHP, COCO InstaVariety | single/multiple persons Motion Discriminator |
| | | | | PennAction, Kinetics | Temporal encoder/decoder |
| | | | | 3DHP, AMASS | single/multiple persons |
| | | | | PoseTrack,3DPW | |
| [49] | 66.1 | RNN | Two | N/A | Simple temporal model |
| [14] | 68.0 | MLP | Two | Kinetics | unsupervised learning with GAN loss |
| [90] | 71.4 | CNN | One | MPII | recurrent refine 3D pose |
| [104] | 80.5 | Resnet50 | One | MPII, LSP, 3DHP | location map,kinematics fitting |
| [86] | 88.77 | Hourglass | One | MPII | trajectory optimization |
| [113] | 101.8 | Hourglass | One | N/A | extract 3D from 2d annotations |
| [219] | 113.01 | CNN | One | PennAction | EM algorithm over heatmaps |
| [159] | 124.97 | CNN | One | N/A | 3D Hog features |
| [60] | 126.5 | CNN | Two | N/A | Height-map |
| [133] | 131.7 | Resnet18 | One | N/A | muti-view supervision |
| **Single Image as Input** | | | | | |
| [214] | 39.9 | Resnet50 | Oen | MPII | part-centric heatmap triplets |
| [68] | 41.1 | Resnet50 | One | LSP-Extended MPII, COCO 3DHP, LSP | Model-fitting in the loop |
| [21] | 42.2 | GCN | Two | N/A | Local-connected GCN |
| [80] | 42.6 | MLP | Two | N/A | multimodal mixture density networks |
| [188] | 43.2 | HRNet | One | MPII | Limb Depth Map |
| [213] | 43.8 | GCN | Two | N/A | first apply GCN to 3D human pose |
| [117] | 44.7 | Hourglass | One | LSP,MPII | use ordinal information between joints |
| [98] | 45.5 | MLP | Two | N/A | simple yet effective baseline |
| [16] | 46.3 | Hourglass | Two | N/A | View synthesis, latent representation. |
| [142] | 46.8 | MLP | Two | CMU-mocap | CVAE model, joint-ordinal relations |
| [194] | 48.0 | Resnet50 | One | MOCA, 3DHP LSP,MPII, COCO | Differential Renderer, IUV map |
| [150] | 48.3 | Resnet50 | One | MPII | additional bone length loss |
| [151] | 49.6 | Resnet50 | One | MPII | Integral of voxel |
| [75] | 50.8 | Resnet50 | One | YTube | bride gap between 3D pose and spatial part maps |
| [174] | 50.9 | MLP | Two | N/A | estimate both 3D pose and cameras |
| [182] | 52.0 | Resnet50 | One | MPII | handle cross-dataset evaluation |
| [112] | 53.2 | Inception v4 | One | MPII | Marginalized voxels |

Table 2.8: Methods on Human36M and the corresponding highlights and performance. Methods based on single frames are at bottom while methods based on videos are at top. Models are trained with subjects 1,3,5,7,8, and tested with subjects 9,11. Unit is in mm. No PA alignment.

| Method (H36M-continue) | MPJPE | Backbone | Stages | Extra data | Highlights |
|---|---|---|---|---|---|
| **Single Image as Input-continue** | | | | | |
| [154] | 54.2 | Resnet50 | One | MPII | winning solution for the H36M challenge |
| [107] | 54.4 | Resnet50 | One | MPII | Multi-person, root estimation |
| [61] | 56.8 | Resnet50 | One | MPII, LSP, COCO LSP-extended CMU-mocap 3DHP, PosePrior | end2end shape estimation |
| [59] | 57.0 | Resnet50 | One | MPII, LSP, COCO | face/hand/body shape estimation |
| [189] | 58.3 | VGG | One | Panoptic Studio STB, COCO D+O | first method to capture the 3D total motion of a target person from a monocular view input. |
| [196] | 58.6 | Hourglass | One | MPII | adversarial training geometric descriptor |
| [114] | 59.9 | Resnet101 | One | UP-3D | conditioned on part segmentation |
| [38] | 60.27 | Resnet50 | One | MPII, COCO | 2d/3D keypoints part, densepose |
| [26] | 61.0 | InceptionV3 | One | N/A | compressed voxels |
| [139] | 61.2 | Resnet50 | One | CMU-mocap | Mask-CNN version 3D pose |
| [215] | 64.9 | Hourglass | One | MPII | 2d/3D bone length constraint |
| [40] | 65.7 | Resnet50 | One | LSP,MPII,H36M | disentangle 2d/3D information |
| [118] | 71.9 | Hourglass | One | MPII | voxel representing 3D pose |
| [120] | 75.9 | Hourglass | One | CMU Mocap, UP-3D SURREAL, MPII, LSP | first one refer to human shape to predict pose with networks |
| [109] | 76.47 | CNN | Two | N/A | euclidean distance matrix |
| [66] | 76.6 | Resnet50 | One | MPII | self-supervised learning |
| [77] | 80.7 | DeeperCut | One | MPII, LSP LSP-extended | The first in the wild dataset with mesh annotation |
| [13] | 82.37 | MLP | Two | CMU-mocap | exemplar-based method |
| [138] | 83.0 | VGG-16 | One | MPII, LSP-extended CMU-mocap, pose prior | extend faster RNN to 3D human pose |
| [160] | 88.39 | CNN | One | N/A | fuse heatmap to get 3D pose |
| [168] | 97.2 | VGG | One | N/A | Adversarial Inverse Graphics |
| [69] | 113.2 | GCN | Two | N/A | direct regress vertex location |
| [137] | 121.2 | CNN | One | N/A | mocap-guided data augmentation |
| [84] | 121.3 | CNN | One | N/A | embedding of poses and images |

Table 2.9: Methods on Human36M and the corresponding highlights and performance. Methods are based on single frame input.

| Method (GPA) | MPJPE | Backbone | Stages | Extra data | Highlights |
|---|---|---|---|---|---|
| **Single Image as Input** | | | | | |
| [182] | 52.0 | Resnet50 | One | MPII | body-center coordinates rotation loss |
| [107] | 53.2 | Resnet50 | One | MPII | root joint estimation |
| [180] | 64.6 | MLP | Two | N/A | geometric affordance |
| [98] | 68.2 | MLP | Two | N/A | simple yet effective baseline |
| [215] | 96.5 | Hourglass | One | MPII | bone length constraint |

Table 2.10: Methods on GPA and the corresponding highlights and performance. Methods are based on single frame input.

| Method (SURREAL) | MPJPE | Backbone | Stages | Extra data | Input | Highlights |
|---|---|---|---|---|---|---|
| [182] | 37.1 | Resnet50 | One | MPII | Image | body-center coordinates<br>rotation loss |
| [107] | 37.2 | Resnet50 | One | MPII | Image | root joint estimation |
| [170] | 49.1 | MLP | Two | N/A | Image | volumetric body shape estimation |
| [169] | 64.4 | MLP | Two | N/A | Video | self-supervised differentiable rendering |

Table 2.11: Methods on SURREAL and the corresponding highlights and performance. Methods are based on single frame input.

| Method (3DHP) | PCK3D | Backbone | Stages | Extra data | Highlights |
|---|---|---|---|---|---|
| **Video Sequence as Input** | | | | | |
| [18] | 93.2 | 1DCNN | Two | N/A | Occlusion Augmentation |
| [64] | 89.3 | RNN | Two | InstaVariety,PoseTrack<br>PennAction, Kinetics<br>3DHP, AMASS,3DPW | Motion Discriminator<br>Temporal encoder/decoder<br>single/multiple persons |
| [87] | 83.6 | MLP | Two | N/A | matrix factorization for<br>sequential 3D human poses |
| [102] | 82.8 | CNN | One | MPII, LSP<br>3DHP, COCO | memory-efficient representation<br>single/multiple persons |
| [23] | 76.7 | Hourglass | One | MPII | generate rotation-valid pose<br>and explore temporal dependence |
| [14] | 64.3 | MLP | Two | Kinetics | unsupervised learning<br>with GAN loss |
| [104] | 79.4 | Resnet50 | One | MPII, LSP, 3DHP | location map,kinematics fitting |
| **Single Image as Input** | | | | | |
| [188] | 93.2 | HRNet | One | MPII | Limb Depth Map |
| [68] | 92.5 | Resnet50 | One | 3DHP, LSP-Extended<br>LSP,MPII, COCO | Model-fitting in the loop |
| [112] | 88.3 | Inception v4 | One | MPII | Marginalized voxels |
| [75] | 84.6 | Resnet50 | One | YTube | bride gap between 3D pose<br>and spatial part maps |
| [182] | 84.3 | Resnet50 | One | MPII | body-center coordinates<br>rotation loss |
| [174] | 82.5 | MLP | Two | N/A | estimate both 3D pose and cameras |
| [66] | 77.5 | Resnet50 | One | MPII | self-supervised learning |
| [194] | 76.9 | Resnet50 | One | MOCA, 3DHP<br>LSP,MPII, COCO | Differential Renderer, IUV map |
| [16] | 75.9 | Hourglass | Two | N/A | View synthesis, latent representation. |
| [214] | 74.3 | Resnet50 | Oen | MPII | part-centric heatmap triplets |
| [21] | 74.0 | GCN | Two | N/A | Local-connected GCN |
| [80] | 72.5 | MLP | Two | N/A | multimodal mixture<br>density networks |
| [117] | 71.9 | Hourglass | One | LSP,MPII | use ordinal information<br>between joints |
| [40] | 70.4 | Resnet50 | One | LSP,MPII,H36M | disentangle 2d/3D information |
| [61] | 72.9 | Resnet50 | One | MPII, LSP, COCO<br>LSP-extended, 3DHP<br>PosePrior, CMU-mocap | end2end shape estimation |

Table 2.12: Methods on 3DHP and the corresponding highlights and performance. Methods are based on single frame input. It is worth noticing the metric for PCK3D is the higher the better.

| Method (3DPW) | PA-MPJPE | Backbone | Stages | Extra data | Highlights |
|---|---|---|---|---|---|
| **Video Sequence as Input** | | | | | |
| [64] | 51.9 | RNN | Two | InstaVariety,PoseTrack<br>PennAction, Kinetics<br>3DHP, AMASS,3DPW | Motion Discriminator<br>Temporal encoder/decoder<br>single/multiple persons |
| [152] | 69.5 | Resnet50 | One | MPII, LSP<br>AICH, Penn Action | skeleton-disentangled<br>representation |
| [18] | 71.8 | 1DCNN | Two | N/A | Occlusion Augmentation |
| [3] | 72.2 | Resnet50 | One | 3DHP, COCO<br>LSP, MPII, Flickr | In the wild human shape<br>reconstruction |
| [62] | 72.6 | 1DCNN | Two | NBA, Penn Action<br>InstaVariety | Two stage mesh estimation |
| [102] | 80.3 | CNN | One | MPII, LSP<br>3DHP, COCO | memory-efficient representation<br>single/multiple persons |
| [23] | 92.3 | Hourglass | One | MPII | generate rotation-valid pose<br>and explore temporal dependence |
| **Single Image as Input** | | | | | |
| [68] | 59.2 | Resnet50 | One | 3DHP, LSP-Extended<br>LSP,MPII, COCO | Model-fitting in the loop |
| [182] | 65.2 | Resnet50 | One | MPII, PASCAL VOC | handle cross-dataset evaluation |
| [61] | 76.7 | Resnet50 | One | MPII, LSP, COCO<br>LSP-extended, 3DHP<br>PosePrior, CMU-mocap | end2end shape estimation |

Table 2.13: Methods on 3DPW and the corresponding highlights and performance. Methods are based on single frame input.

# Chapter 3

# Predicting Camera Viewpoint Improves Cross-dataset Generalization for 3D Human Pose Estimation

## 3.1 Introduction

A large swath of computer vision research increasingly operates in playing field which is swayed by the quantity and quality of annotated training data available for a particular task. How well do you know your data? Fig 3.1 presents a sampling images from 5 popular datasets used for training models for 3D human pose estimation (Human3.6M [52], GPA [180], SURREAL [171], 3DPW [173] , 3DHP [101]). We ask the reader to consider the game of "Name That Dataset" in homage to Torralba *et al.* [162]. Can you guess which dataset each image belongs to? More importantly, if we train a model on the Human3.6M dataset (at Fig 3.1 left) how well would you expect it to perform on each of the images depicted?

Each of these datasets were collected using different mocap systems (VICON, The Capture,

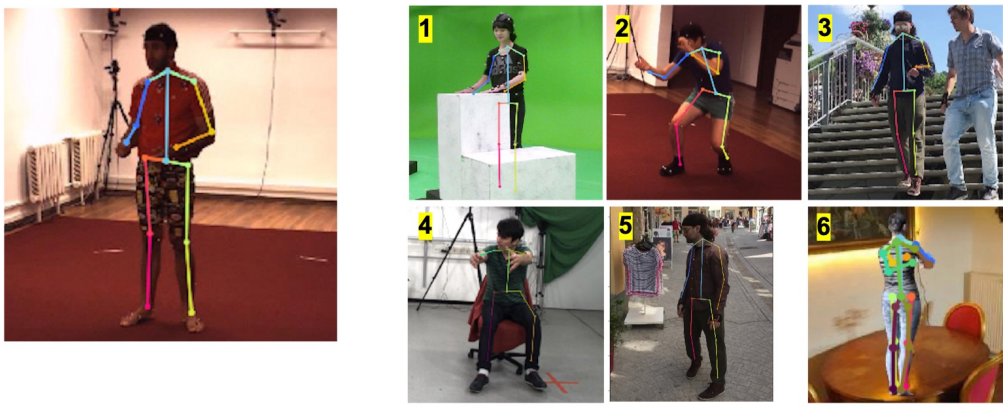



Figure 3.1: In this chapter we consider the problem of dataset bias and cross-dataset generalization. Can you guess which human pose dataset each image on the right comes from? If we train a model on H36M data (left) can you predict which image has the lowest/highest 3D pose prediction error? (answer key below)[1]

IMU), different cameras (Kinect, commercial synchronized cameras, phone), and collected in different environments (controlled lab environment, marker-less in the wild environment, or synthetic images) with varying camera viewpoint and pose distributions (see Fig 3.3). These datasets contain further variations in body sizes, camera intrinsic and extrinsic parameters, body and background appearance. Despite the obvious presence of such systematic differences, these variables and their subsequent effect on performance have yet to be carefully analyzed.

In this chapter, we study the generalization of 3D pose models across multiple datasets and propose an auxiliary prediction task: estimating the relative rotation between camera viewing direction and a body-centered coordinate system defined by the orientation of the torso. This task serves to significantly improve cross-dataset generalization. Ground-truth for our proposed camera viewpoint task can be derived for existing 3D pose datasets without requiring additional labels. We train off-the shelf models [107, 215] which estimate the camera-relative 3D pose, augmented with a viewpoint prediction branch. In our experiments, we show our approach outperforms the state-of-the-art PoseNet [107] and [215] baseline by

[1]Answer key: Metric: MPJPE, the lower the better. 1) GPA: 69.7 mm 2) H36M: 29.2 mm, 3) 3DPW, 71.2 mm, 4) 3DHP 107.7 mm, 5) 3DPW 66.2 mm, 6) SURREAL 83.4 mm, H36M image performs best while 3DHP image performs worst.

a large margin across 5 different 3D pose datasets. Perhaps even more startling is that the addition of this auxiliary task results in significant improvement in cross-dataset test performance. This simple approach increases robustness of the model and, to our knowledge, is the first work that systematically confronts the problem of dataset bias in 3D human pose estimation.

To summarize, our main contributions are:

• We analyze the differences among contemporary 3D human pose estimation datasets and characterize the distribution and diversity of viewpoint and body-centered pose.

• We propose the novel use of camera viewpoint prediction as an auxiliary task that systematically improves model generalization by limiting overfitting to common viewpoints and can be directly calculated from commonly available joint coordinate ground-truth.

• We experimentally demonstrate the effectiveness of the viewpoint prediction branch in improving cross-dataset 3D human pose estimation over two popular baseline and achieve state-of-the-art performance on five datasets.

## 3.2 Related Work

**Cross-Dataset Generalization and Evaluation** 3D human pose estimation from monocular imagery has attracted significant attention due to its potential utility in applications such as motion retargeting [172], gaming, sports analysis, and health care [97]. Recent methods are typically based on deep neural network architectures [17, 70, 71, 98, 107, 118, 122, 151, 181, 184, 215, 185, 179, 183, 72, 72] trained on one of a few large scale, publicly available datasets. Among these are [98, 118, 151] evaluated on H36M, [101, 215] work on both H36M

[52] and 3DHP [101], [173, 164] work on TOTALCAPTURE [164] and 3DPW[173], [180] work on the GPA dataset [180]. [171] works on both SURREAL [171] and H36M [52] dataset.

Given the powerful capabilities of CNNs to overfit to specific data, we are inspired to revisit the work of [162], which presented a comparative study of popular object recognition datasets with the goals of improving dataset collection and evaluation protocols. Recently, [76] observed characteristic biases present in commonly used depth estimation datasets and proposed scale invariant training objectives to enable mixing multiple, otherwise incompatible datasets. [225] introduced the first large-scale, multi-view unbiased hand pose dataset as training set to improve performance when testing on other dataset. Instead of proposing yet another dataset or resorting to domain adaptation approaches (see e.g., [177]), we focus on identifying systematic biases in existing data and identifying generic methods to prevent overfitting in 3D pose estimation.

**Coordinate Frames for 3D Human Pose** In typical datasets, gold-standard 3D pose is collected with motion capture systems [52, 145, 164, 180] and used to define ground-truth 3D pose relative one or more calibrated RGB camera coordinate systems [52, 173, 101, 171, 180]. To generate regression targets for use in training and evaluation, it is typical to predict the *relative* 3D pose and express the joint positions relative to a specified root joint such as the pelvis (see e.g.,[107, 151]). We argue that camera viewpoint is an important component of the experimental design which is often overlooked and explore using a body-centered coordinate system which is rotated relative to the camera frame.

This notion of view-point invariant prediction has been explored in the context of 3D object shape estimation [20, 37, 105, 135, 143, 156, 167, 195] where many works have predicted shape in either an object-centered or camera-centered coordinate frame [143, 157, 211]. Closer to our task is the 3D hand pose estimator of [224] which separately estimated the viewpoint and pose (in canonical hand-centered coordinates similar to ours) and then combine the two

to yield the final pose in the camera coordinate frame. However, we note that predicting canonical pose directly from image features is difficult for highly articulated objects (indeed subsequent work on hand pose, e.g. [53], abandoned the canonical frame approach). Our use of body-centered coordinate frames differs in that we only use them as a auxiliary training task that improves prediction of camera-centered pose.

**3D Human Pose Estimation** With the recent development of deep neural networks (CNNs), there are significant improvements on 3D human pose estimation [41, 98, 118, 190]. Many of them try to tackle in-the-wild images. [215] proposes to add bone length constraint to generalize their methods to in the wild image. [139] seeks to pose anchors as classification template and refine the prediction with further regression loss. [41] propose a a new disentangled hidden space encoding of explicit 2D and 3D features for monocular 3D human pose estimation that shows high accuracy and generalizes well to in-the-wild scenes, however, they do not evaluate its capacity on indoor cross-dataset generalization. To the best of our knowledge, our work is the first to exploit cross-dataset task not only towards in-the-wild generalization but also across different indoor datasets.

**Multi-task Training** There have has been a wide variety of work in training deep CNNs to perform multiple tasks, for example: joint detection, classification, and segmentation [45], joint surface normal, depth, and semantic segmentation [67], joint face detection, keypoint, head orientation and attributes [129]. Such work typically focuses on the benefits (accuracy and computation) of jointly training a single model for two or more related tasks. For example, predicting face viewpoint has been shown to improve face recognition [200]. Our approach to improving generalization differs in that we train models to perform two tasks (viewpoint and body pose) but discard viewpoint predictions at test time and only utilize pose. In this sense our model is more closely related to work on "deeply-supervised" nets [78, 192] which trains using losses associated with auxiliary branches that are not used at test time.

| Dataset | H36M | GPA | SURREAL | 3DPW | 3DHP |
|---|---|---|---|---|---|
| Year | 2014 | 2019 | 2017 | 2018 | 2017 |
| Imaging Space | 1000 × 1002 | 1920 × 1080 | 320 × 240 | 1920 × 1080 | 2048 × 2048 or 1920 × 1080 |
| Camera Distance | 5.2 ± 0.8 | 5.1 ± 1.2 | 8.0 ± 1.0 | 3.5 ± 0.7 | 3.8 ± 0.8 |
| Camera Height | 1.6 ± 0.05 | 1.0 ± 0.3 | 0.9 ± 0.1 | 0.6 ± 0.8 | 0.8 ± 0.4 |
| Focal Length | 1146.8 ± 2.0 | 1172.4 ± 121.3 | 600 ± 0 | 1962.2 ± 1.5 | 1497.88 ± 2.8 |
| No. of Joints | 38 | 34 | 24 | 24 | 28 or 17 |
| No. of Cameras | 4 | 5 | 1 | 1 | 14 |
| No. of Subjects | 11 | 13 | 145 | 18 | 8 |
| Bone Length | 3.9 ± 0.1 | 3.7 ± 0.2 | 3.7 ± 0.2 | 3.7 ± 0.1 | 3.7 ± 0.1 |
| GT source | VICON | VICON | Rendering | SMPL | The Captury |
| No. Train Images | 311,951 | 222,514 | 867,140 | 22,375 | 366,997 |
| No. Test Images | 109,764 | 82,378 | 507 | 35,515 | 2,875 |

Table 3.1: Comparison of existing datasets commonly used for training and evaluating 3D human pose estimation methods. We calculate the mean and std of camera distance, camera height, focal length, bone length from training set. Focal length is in mm while the others are in unit meters. 3DHP has two kinds of cameras and the training set provide 28 joints annotation while test set provide 17 joints annotation.

## 3.3 Variation in 3D Human Pose Datasets

We begin with a systematic study of the differences and biases across 3D pose datasets. We selected three well established datasets Human3.6m (H36M), MPI-inf-3Dhp (3DHP), SURREAL, as well as two more recent datasets 3DPW and GPA for analysis. These are large-scale datasets with a wide variety of characteristics in terms of capture technology, appearance (in-the-wild,in-the-lab,synthetic) and content (range of body sizes, poses, viewpoints, clothing, occlusion and human-scene interaction). In this chapter, we focus on characterizing variation in geometric quantities (pose and viewpoint) which can be readily quantified (compared to, e.g., lighting and clothing).

We list some essential statistics from 5 datasets in Table 3.1. For these datasets, gold-standard 3D pose is collected with motion capture systems [52, 145, 164, 180] and used to define ground-truth 3D pose relative one or more calibrated RGB camera coordinate systems [52, 173, 101, 171, 180]. To generate regression targets for use in training and

evaluation, it is typical to predict the *relative* 3D pose (see e.g.,[107, 151]) and express the joint positions relative to a specified root joint (typically the pelvis) and crop/scale the input image accordingly. This pre-processing serves to largely "normalize away" dataset differences in camera intrinsic parameters and camera distance shown in Table 3.1. However, it does not address camera orientation.

To characterize the remaining variability, we factor the camera-relative pose into camera viewpoint (the position of the camera relative to a canonical body-centered coordinate frame defined by the orientation of the person's torso) and the pose relative to this body-centered coordinate frame.

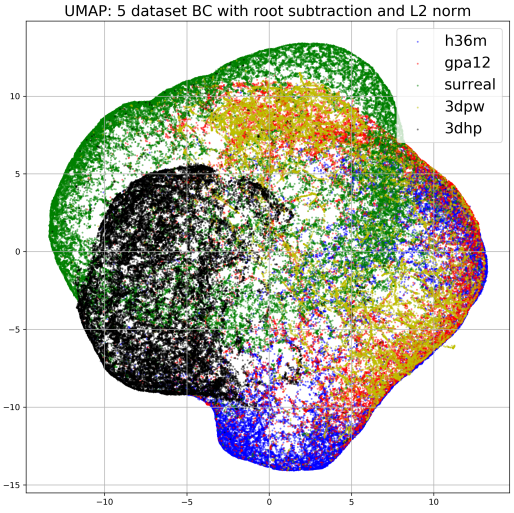
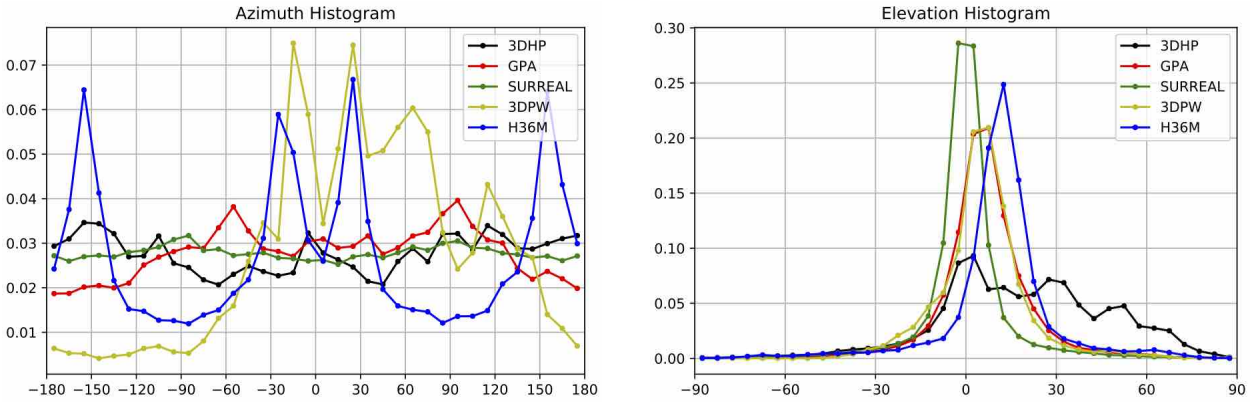



Figure 3.2: Distribution of view-independent body-centered pose, visualized as a 2D embedding produced with UMAP [100]

**Computing Body-centered Coordinate Frames** To define a viewpoint-independent pose, we need to specify a canonical body-centered coordinate frame. As shown in Fig 3.9a, we take the origin to be the camera-centered coordinates of root joint (pelvis) $p_p = (x_p, y_p, z_p)$ and the orientation is defined by the plane spanned by $p_p$, the left shoulder $p_l$ and the right shoulder $p_r$. Given these joint positions, we can compute an orthogonal frame consisting of the front direction $f$, up direction $u$ and right direction $r$ are




(a) Viewpoint Azimuth (b) Viewpoint Elevation

Figure 3.3: Distribution of camera viewpoints relative to the human subject. We show the distribution of camera azimuth $(-180°, 180°)$ and elevation $(-90°, 90°)$ for 50k poses sampled from each representative dataset (**H36M**, **GPA**, **SURREAL**, **3DPW**, **3DHP**).

defined as:

$$u = (p_l + p_r)/2 - p_p \tag{3.1}$$

$$f = (p_l - p_p) \times (p_r - p_p) \tag{3.2}$$

$$r = f \times u \tag{3.3}$$

The rotation between the body-centered frame and the camera frame is then given by the matrix $R = -[r, u, f]$. We find it useful to represent rotations using unit quaternions (as have others, e.g. [172, 166]). The corresponding unit quaternion representing $R$ has components:

$$q = \frac{1}{4q_0}[4q_0^2, u_2 - f_1, f_0 - r_2, r_1 - u_0], \qquad q_0 = \sqrt{(1 - r_0 - u_1 - f_2)} \tag{3.4}$$

**Distribution of Camera Viewpoints** Fig 3.3 shows histograms capturing the distribution of camera viewing direction in terms of azimuth (Fig 3.3a) and elevation (Fig 3.3b) relative to the body-centered coordinate system for 50k sample poses from each of the 5 datasets.

We observe **H36M** has a wide range of view direction over azimuth with four distinct peaks

(−30 degree, 30 degree, −160 degree, 160 degree), it shows during the capture session subjects are always facing towards or facing away the control center while the four RGB cameras captured from four corners. H36M has a clear bias towards elevation above 0; **GPA** is more spread over azimuth compared with H36M, most of the views range from −60 degree to 90 degree; **SURREAL** synthetically sampled camera positions with a uniform distribution over azimuth, and also have a uniform

distribution over elevation. The viewpoint bias for **3DPW** arises naturally from filming people in-the-wild from a handheld or tripod mounted camera roughly the same height as the subject. Of the non-synthetic datasets, **3DHP** is the most uniform spread over azimuth and includes a wider range of positive elevations, a result of utilizing cameras mounted at multiple heights including the ceiling.

These differences are further highlighted in Fig 3.9 which shows the joint distribution of camera views and reveals the source of non-uniformity of the azmuthal distribution for 3DHP and H36M due to subjects tending to face a canonical direction while performing some actions. For example, in H36M in Fig 3.9b, actions in which the subject lean over or lie down (extreme elevations) only happen at particular azimuths. Similarly, in 3DHP (Fig 3.9f), the 14 camera locations are visible as dense clusters at specific azimuths indicating a significant subset of the data in which the subject was facing in a canonical direction relative to the camera constellation.

**Distribution of Pose** To characterize the remaining variability in pose after the viewpoint is factored out, we used the coordinates of 14 joints common to all datasets expressed in the body-centered coordinate frame. We also scaled the body-centered joint locations to a common skeleton size (removing variation in bone length shown in Table 1). To visualize the resulting high-dimensional data distribution, we utilized UMAP [100] to perform a non-linear embedding into 2D. Figure 3.2 shows the resulting distributions which show a substantial

degree of overlap. For comparison, please see the figure 3.8 which show embeddings of the same data when bone length and/or viewpoint are not factored out.

We also trained a multi-layer perceptron to predict which dataset a given body-relative pose came from. It had an average test accuracy of 20% providing further evidence of relatively little bias in the distribution of poses across datasets once viewpoint and body size are factored out.

## 3.4 Learning Pose and Viewpoint Prediction

To overcome biases in viewpoint across datasets, we propose to use viewpoint prediction as an auxiliary task to regularize the training of standard camera-centered pose estimation models.

### 3.4.1 Baseline architecture

Our baseline model [107, 215] consists of two parts: the first ResNet [46] backbone which takes in images patches cropped around the human; followed by the second part which takes the resulting feature map and upsamples it using three consecutive deconvolutional layers with batch normalization and ReLU. A 1-by-1 convolution is applied to the upsampled feature map to produce the 3D heatmaps for each joint location. The soft-argmax [151] operation is used to extract the 2D image coordinates $(\hat{x}_j, \hat{y}_j)$ of each joint $j$ within the crop, and the root-relative depth $\hat{y}_j$. At test time, we can convert this prediction into into a 3D metric joint location $p_j = (x_j, y_j, z_j)$ using the crop bounding box, an estimate of the root joint depth or skeleton size, and the camera intrinsic parameters.

The loss function of the coordinate branch is the $L1$ distance between the estimated and

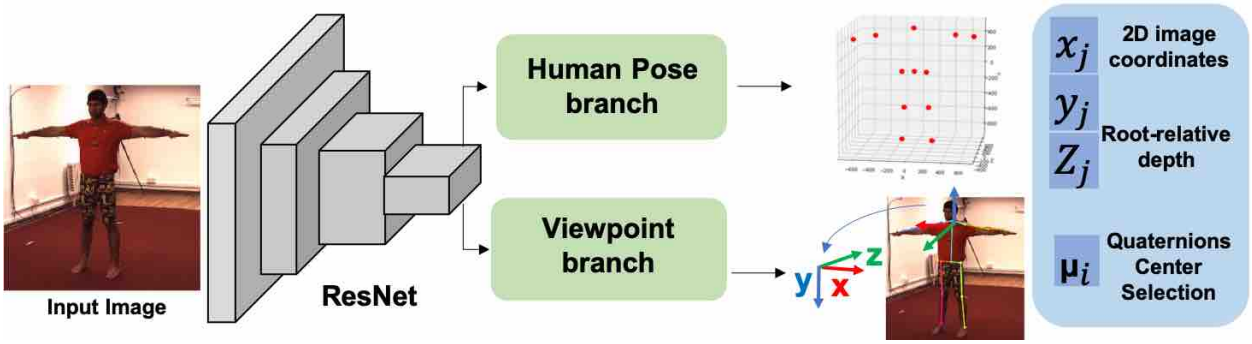



Figure 3.4: Flowchart of our model. We augment a model which predicts camera-centered 3D pose using the **human pose branch** with an additional **viewpoint branch** that selections among a set of quantized camera view directions.

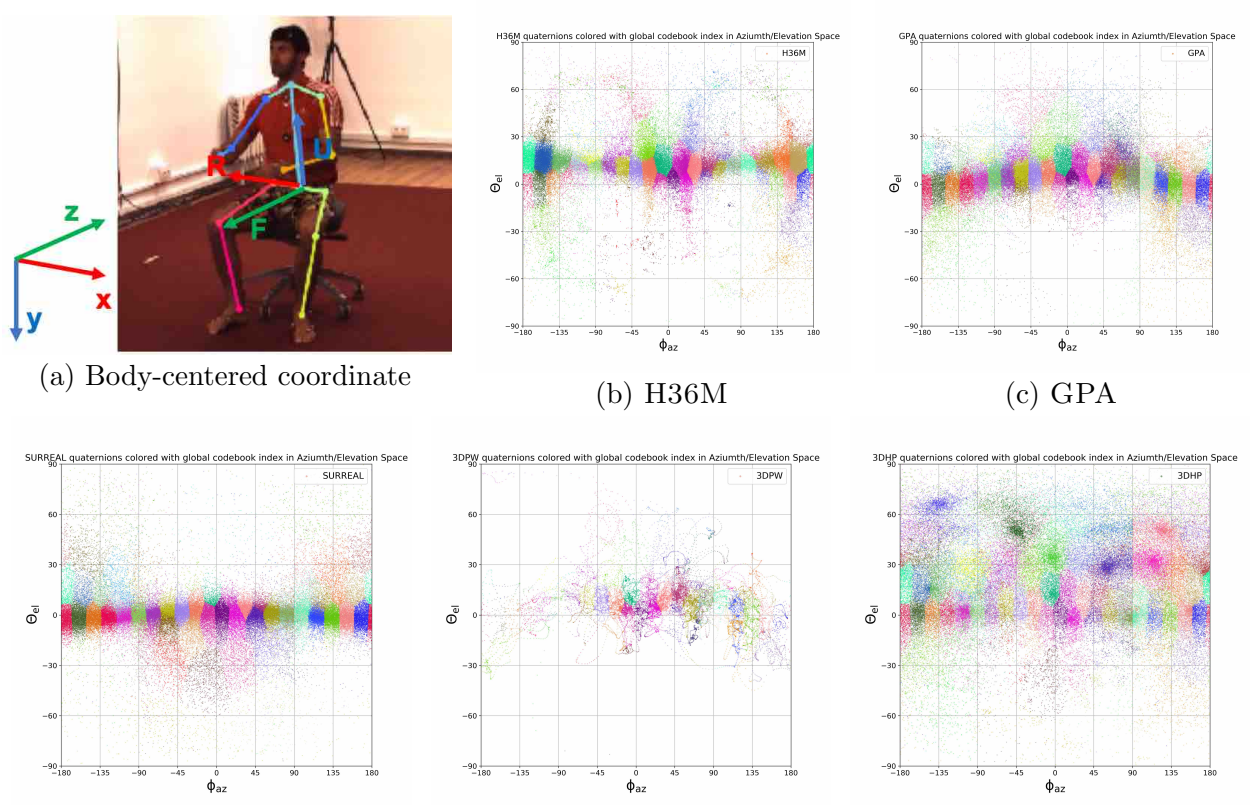
groud-truth coordinates.

$$\mathcal{L}_{pose} = \frac{1}{J}\sum_{j=1}^{J} ||p_j - p_j^*||_1 \tag{3.5}$$

### 3.4.2 Predicting the camera viewpoint

To predict the camera viewpoint relative to the body-centered coordinate frame we considered three approaches: (i) direct regression of $q$, (ii) quantizing the space or rotations and performing k-way classification, and (iii) a combined approach of first predicting a quantized rotation followed by regressing the residual from the cluster center. In our experiments, we found that the classification-based loss yields less accurate coordinate frame predictions but yielded the largest improvements in the pose prediction branch (see Table 3.4).

To quantize the space of rotations, we use k-means to cluster the quaternions into k=100 clusters. The clusters are computed from training data of a single dataset (local clusters) or from all five datasets (global clusters). We visualize the global cluster centers in azimuth and elevation space in Fig 3.9 b-f, as well as randomly sampled quaternions from H36M, GPA, SURREAL, 3DPW and 3DHP datasets.




(a) Body-centered coordinate (b) H36M (c) GPA

(d) SURREAL (e) 3DPW (f) 3DHP

Figure 3.5: **a**: Illustration of our body-centered coordinate frame (up vector, right vector and front vector) relative to a camera-centered coordinate frame. **b-f**: Camera viewpoint distribution of the 5 datasets color by quaternion cluster index. Quaternions (rotation between body-centered and camera frame) are sampled from training sets and clustered using k-means. They are also visualized in azimuth / elevation space following Fig 3.3.

To regress the quaternion $q$ we simply add a branch to our base pose prediction model consisting of a 1x1 convolutional layer to reduce the feature dimension to 4 followed by global average pooling and normalization to yield a unit 4-vector. We train this variant using a standard squared-Euclidan loss on target $q^*$. For classification, we use the same prediction $q$ but compute the probability it belongs to the correct cluster using a softmax to get a distribution over cluster assignments:

$$p(c|q) = \frac{\exp(-\mu_c^T q)}{\sum_{i=1}^{k} \exp(-\mu_i^T q)} \tag{3.6}$$

where $\{\mu_1, \mu_2, \ldots, \mu_k\}$ are the quaternions corresponding to cluster centers computed by

k-means. We use the negative log-likelihood as the training loss,

$$\mathcal{L}_q = -log(p(c^*|q)) \tag{3.7}$$

where $c^*$ is the viewpoint bin that the training example was assigned during clustering. Our final loss consists of both quaternion and pose terms: $\mathcal{L} = \lambda\mathcal{L}_q + \mathcal{L}_{pose}$.

## 3.5 Experiments

**Data and evaluation metric.** To reduce the redundancy of the training images (30 fps video gives lots of duplicated images for network training), we down sample 3DHP, SURREAL to 5 fps. Following [107, 215], we sample H36M to 10 fps, and use the protocol 2 (subject 1,3,5,7,8 for training and subject 9,11 for testing, and here we report MPJPE over samples instead of over classes, which is a harder setting based on our experience) for evaluation. As GPA is designed as monocular image 3D human pose estimation, which is already sampled, we follow [180] and directly use the released set. Number of images in train set and test set is shown in Table 3.1. In addition, we use the MPII dataset [2], a large scale in-the-wild human pose dataset for training a more robust pose model. It contains 25k training images and 2,957 validation images. We use two metrics, first is mean per joint position error (MPJPE), which is calculated between predicted pose and ground truth pose. The second one is PCK3D [101], which is the accuracy of joint prediction (threshold on MPJPE with 150mm).

**Implementation Details.** As different datasets have diverse joint configuration, we select a subset of 14 joints that all datasets share to eliminate the bias introduced by different number of joints during training.

We normalize the z value from $(-z_{max}, +z_{max})$ to $(0, 63)$ for integral regression. $z_{max}$ is 2400

| | | MPJPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|---|
| | Testing \Training | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Baseline | H36M | **53.2** | 110.5 | 107.1 | 125.1 | 108.4 |
| | GPA | 105.2 | **53.9** | 86.8 | 111.7 | 90.5 |
| | SURREAL | 118.6 | 103.2 | **37.2** | 120.8 | 108.2 |
| | 3DPW | 108.7 | 116.4 | 114.2 | **100.6** | 113.3 |
| | 3DHP | 111.8 | 123.9 | 120.3 | 139.7 | **91.9** |
| Our Method | H36M | **52.0** | 102.5 | 103.3 | 124.2 | 95.6 |
| | GPA | 98.3 | **53.3** | 85.6 | 110.2 | 91.3 |
| | SURREAL | 114.0 | 101.2 | **37.1** | 113.8 | 107.2 |
| | 3DPW | 109.5 | 112.0 | 112.2 | **89.7** | 105.9 |
| | 3DHP | 111.9 | 119.7 | 118.2 | 136.0 | **90.3** |
| Same-Dataset Error Reduction ↓ | | 1.2 | 0.6 | 0.1 | 10.9 | 1.5 |
| Cross-Dataset Error Reduction ↓ | | 10.6 | 18.6 | 9.1 | 13.1 | 20.4 |

Table 3.2: Baseline cross-dataset test error and error reduction from the addition of our proposed quaternion loss. Bold indicates the best performing model on each the test set (rows). Blue color indicates test set which saw greatest error reduction. See appendix for corresponding tables of PCK and Procrustese aligned MPJPE.

mm based all 5 set. We use PyTorch to implement our network. The ResNet-50 [46] backbone is initialized using the pre-trained weights on the ImageNet dataset. We use the Adam [63] optimizer with a mini-batch size of 128. The initial learning rate is set to $1 \times 10^{-3}$ and reduced by a factor of 10 at the 17th epoch, we train 25 epochs for each of the dataset. We use $256 \times 256$ as the size of the input image of our network. We perform data augmentation including rotation, horizontal flip, color jittering and synthetic occlusion following [107]. We set $\lambda$ to 0.5 for the quaternion loss which is validated on 3DPW validation set.

### 3.5.1 Cross-dataset evaluation

We list the cross-dataset baseline and our improved results in Table 3.2. The bold numbers indicate the best performing model on the test set. As expected, the best performance occurs when the model is trained and evaluated on the same set. The numbers marked with blue color indicate the test set where the error reduction is most significant, using our proposed quaternion loss.

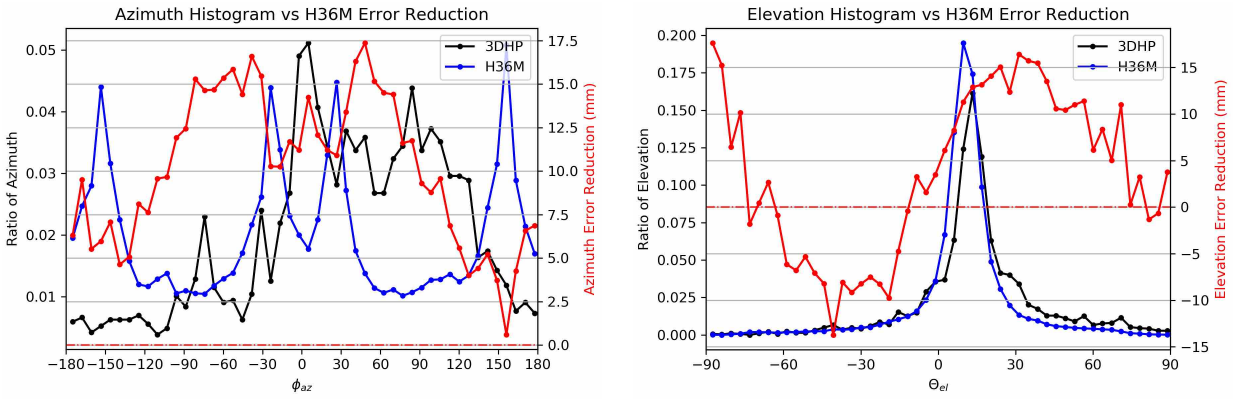



Figure 3.6: We visualize viewpoint distributions for train (3DHP) and test (H36M) overlayed with the reduction in pose prediction error relative to baseline

**Training on H36M.** Adding the quaternion loss reduces total cross-dataset error by 10.6 mm (MPJPE), while the same-dataset error reduction is 1.2 mm (MPJPE). This may be explained by the error on H36M already being low. The largest error reduction is on GPA (6.9 mm) which we attribute to de-biasing the azimuth distribution difference as shown in Fig 3.3a.

**Training on GPA.** The total cross-dataset error reduction is 18.6 mm (MPJPE), and the same data error reduction is 0.6 mm (MPJPE). We attribute this to the bias during capture [180]: the coverage of camera viewing directions is centered in the range of $-60$ to 90 degrees azimuth (as in Fig 3.3a). The largest cross-data set error reduction occurs for H36M, with 8.0 mm. This further demonstrates that the view direction distribution is largely different from H36M.

**Training on SURREAL.** Adding the quaternion loss reduces the cross-dataset error by 9.1 mm (MPJPE), while the same-dataset error reduction is 0.1 mm (MPJPE). We attribute this to the fact that viewpoint distribution on SURREAL itself is already uniform as in Fig 3.3a. We can see distribution over azimuths is quite uniform. Thus adding more supervision in the form of quaternion loss helps little. The most error reduction (2.0mm) is observed on

| | MPJPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|
| Metric \Training Set | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Same-Dataset Error Reduction ↓ | 0.6 | 4.2 | 0.2 | 7.6 | 1.2 |
| Cross-Dataset Error Reduction ↓ | 2.4 | 12.3 | 1.9 | 10.1 | 9.3 |

Table 3.3: Retraining the model of Zhou *et al.* [215] using our viewpoint prediction loss yields also shows significant decrease in prediction error, demonstrating the generality of our finding. See appendix for full table of numerical results.

| Datasets | Baseline | C | R | C+R | C+local cluster | C+cannonical pose |
|---|---|---|---|---|---|---|
| 3DPW (MPJPE (mm)) | 100.6 | 89.7 | 94.0 | 93.2 | 93.1 | 100.3 |

Table 3.4: Ablation analysis: we compare the performance of our proposed camera view-point loss using classification (C), regression (R), using both (C+R); using per-dataset clusterings (local) rather than the global clustering; and adding a third branch which also predicts pose in canonical body-centered coordinates.

3DPW. We attribute this to the fact that 3DPW is strongly biased dataset in terms of view direction, and the quaternion loss helps reduce the view difference between SURREAL and 3DPW.

**Training on 3DPW.** The error is reduced by 10.9 mm (MPJPE) on itself (also the most error reduction one with model trained on 3DPW), and the cross-dataset error reduction is 13.1 mm (MPJPE). From the Fig 3.3a we can see, in terms of azimuth, 3DPW has a strong bias towards $-30$ degree to 60 degree. As during capture, the subject is always facing towards the camera to make it easier for association between the subject (there are multiply persons in crowded scene) and IMU sensors, this bias seems inevitable and quaternion loss is helpful for this kind of in the wild dataset to reduce view direction bias. It is also verified in 3DHP, where half of the test set is in the wild, and have view direction bias.

**Training on 3DHP.** Adding the quaternion loss reduces the total cross-dataset error by 20.4 mm, while the same-dataset error reduction is 1.5 mm (MPJPE). During the capture, 3DHP capture images from a wide range of viewpoints. We can see from the Fig 3.3 that the azimuth of 3DHP is the most uniformly distributed of the real datasets. Thus treating it as training set will enable the network to be robust to view direction. We also calculate

| | MPJPE↓: lower is better | | | | | PCK3D↑: higher is better | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | H36M | GPA | SURREAL | 3DPW | 3DHP | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Mehta [101] | 72.9 | - | - | - | - | - | - | - | - | 64.7 |
| Zhou [215] | 64.9 | 96.5 | - | - | - | - | 82.9 | - | - | 72.5 |
| Arnab[3] | 77.8 | - | - | - | - | - | - | - | - | - |
| Kanazawa [61] | 88.0 | - | - | - | 124.2 | - | - | - | - | 72.9 |
| Kanazawa [62] | - | - | - | 127.1 | - | - | - | - | **86.4*** | - |
| Moon [107] | 54.3 | - | - | - | - | - | - | - | - | - |
| Kolotouros [69] | 78.0 | - | - | - | - | - | - | - | - | - |
| Tung[169] | 98.4 | - | 64.4* | - | - | - | - | - | - | - |
| Varol[170] | **51.6*** | - | 49.1 | - | - | - | - | - | - | - |
| Habibie [41] | 65.7 | - | - | - | 91.0 | - | - | - | - | 82.0 |
| Yu [201] | 59.1 | - | - | - | - | - | - | - | - | - |
| Ours | 52.0 | **53.3** | **37.1** | **89.7** | **90.3** | **96.0** | **96.8** | **97.3** | 84.6 | **84.3** |

Table 3.5: Comparison to state-of-the-art performance. There are many missing entries, indicating how infrequent it is to perform multi-dataset evaluation. Our model provides a new state-of-the art baseline across all 5 datasets and can serve as a reference for future work. * denotes training using extra data or annotations (e.g. segmentation). Underline denotes the second best results.

error reduction conditioned on azimuth and elevation on the H36M test set (Fig 3.6). The blue/black line is azimuth and elevation histogram distribution for H36M/3DHP training sets while the red line shows relative error reduction for H36M. We can see the error is reduced more where H36M has fewer views relative to 3DHP.

### 3.5.2 Effect of Model Architecture and Loss Functions

To demonstrate the generalization of our approach to other models, we also added a viewpoint prediction branch to the model of [215] which utilizes a different model architecture. We observe similar results in terms of improved generalization (see Table 3.3 and appendix). We note that while our primary baseline model [107] uses camera intrinsic parameters to back-project, [215] utilizes an average bone-length estimate from the training set which results in higher prediction errors across datasets.

**Ablation study** To explore whether our methods are robust to different k-means initialization, we repeat k-means 4 times and report performance on 3DPW. We find the range of the

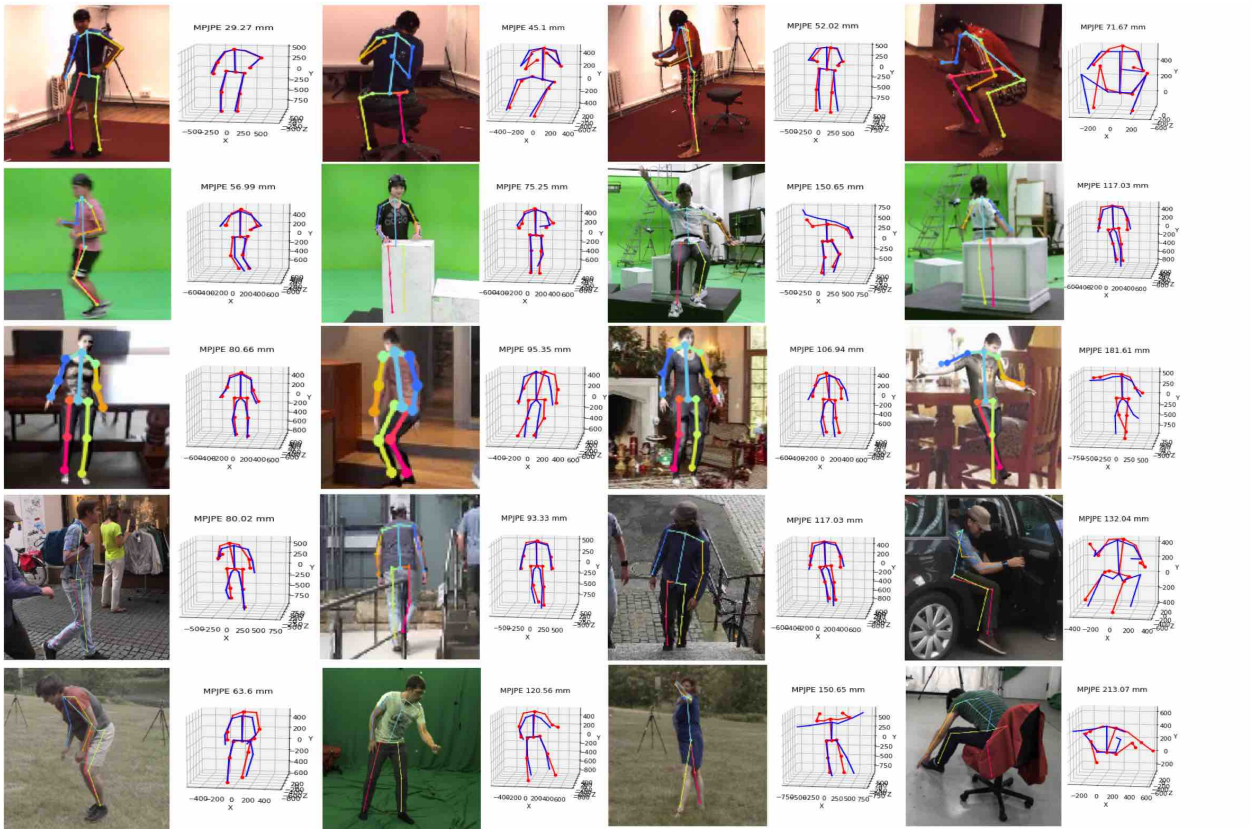



Figure 3.7: Model predictiosn on 5 datasets from model trained on Human3.6M dataset. The 2d joints are overlaid with the original image, while the 3D prediction (red) is overlaid with 3D ground truth (blue). 3D prediction is **visualized in body-centered coordinate** rotated by the relative rotation between ground truth camera-centered coordinate and body-centered coordinate. From top to bottom are H36M, GPA, SURREAL, 3DPW and 3DHP datasets. We rank the images from left to right in order of increasing MPJPE.

MPJPE is within $90 \pm 0.4$ ([89.9, 89.6, 90.2, 89.7]) mm. We also vary the number of clusters to select the best $k \in \{10, 24, 50, 100, 200, 500\}$, with corresponding errors [93.0, 95.2, 92.3, 89.7, 93.0,93.2]. We find k=100 is the best number with at most 6 mm reduction compared to k=24. In Table 3.4, the error of global clusters is 3.4 mm error less than local, per-dataset clusters, demonstrating training on global clusters is better than local clusters which are biased towards the training set view distribution. In terms of choice for quaternion regression, k-way classification reduced error by 4.3 mm compared to regression. While utilizing both classification and regression losses gives error than regression only.

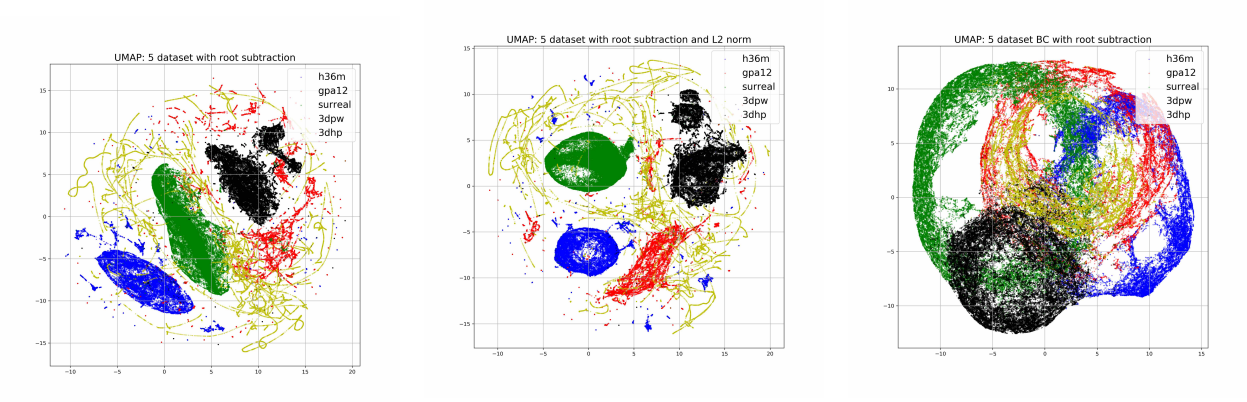
Finally, we also consider adding a third branch and loss function to the model which also predicts the 3D pose in the body-centered coordinate system. This is related to the hand

pose model of [224], although we don't use this prediction of canonical pose at test time. This variant performs global pooling on the ResNet feature map after upsampling followed by a two layer MLP that predicts the viewpoint $q$ and canonical pose. When training with this additional branch we find the camera-centered pose predictions show no improvement over baseline (Table 3.4). We also observe that the canonical pose predictions have higher error than the camera-centered predictions which is natural since the the model can't directly exploit the direct correspondence between the 2D keypoint locations and the 3D joint locations.

### 3.5.3 Comparison with state-of-the-art performance

Table 3.5 compares the proposed approach with the state-of-the-art performance on all 5 datasets. Note that our method is the first to evaluate 3D human pose estimation on the five representative datasets reporting both MPJPE and PCK3D, which fills in some blanks and serves as a useful baseline for future work. As can be seen, our method achieves state-of-the-art performance on H36M/GPA/SURREAL/3DPW/3DHP datasets in terms of MPJPE. While [62] uses additional data (both H36M and 3DHP, and LSP together with MPII) to train, they have slightly better performance on 3DHP in terms of PCK3D.

**Qualitative Results:** We visualize the prediction on the 5 datasets with model trained on H36M using our proposed method in Fig 3.7. The 2d joint prediction is overlaid with cropped images while the 3D joint prediction is visualized in our proposed body-centered coordinates. From top to bottom are H36M, GPA, SURREAL, 3DPW and 3DHP datasets. We display the images from left to right in ascending order by MPJPE.




(a) UMAP WITH ONLY ROOT-SUBTRACTION

(b) UMAP WITH ROOT-SUBTRACTION AND L2 NORMALIZATION

(c) UMAP BODY-CENTERED COORDINATES WITH ONLY ROOT-SUBTRACTION

Figure 3.8: Distribution of view-dependent, view-independent body-centered pose, visualized as a 2D embedding produced with UMAP [100].

## 3.6 UMAP Visualization

We visualize the UMAP [100] embedding of view-dependent coordinate (root-relate coordinate) of H36M [52], GPA [180] , SURREAL [171], 3DPW [173] and 3DHP [101] datasets in Fig 3.8a. We further normalize out skeleton size and visualize in Fig 3.8b. To compare with view-independent coordinate (body-center coordinate), we visualize them before L2 normalization in Fig 3.8c. We can see the body-centered, size normalized pose distribution (main chapter) shows much higher overlap across datasets while the root-relative coordinates implicitly which encode camera orientation provide distinguishable information (dataset bias).

## 3.7 Alternative Model with our quaternion loss

We provide PMPJPE in Table 3.6 and PCK3D in Table 3.7 to demonstrate the effectiveness of adding quaternion loss to PoseNet [107]. To demonstrate the utility of our quaternion loss on other models, we also show results based on retraining the model of [215] in Table 3.8 with MPJPE metric.

| | | PA-MPJPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|---|
| | Testing \Training | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Baseline | H36M | **43.4** | 75.0 | 69.6 | 91.3 | 75.0 |
| | GPA | 75.4 | **41.7** | 66.3 | 84.4 | 70.2 |
| | SURREAL | 76.5 | 73.5 | **31.8** | 85.8 | 77.9 |
| | 3DPW | 68.0 | 66.9 | 64.3 | **68.7** | 68.1 |
| | 3DHP | 88.5 | 91.2 | 86.9 | 111.3 | **71.4** |
| Our Method | H36M | **42.5** | 69.5 | 67.5 | 91.4 | 72.6 |
| | GPA | 71.4 | **40.9** | 65.6 | 81.4 | 70.6 |
| | SURREAL | 75.9 | 71.7 | **31.7** | 82.1 | 76.9 |
| | 3DPW | 68.3 | 65.1 | 63.8 | **65.2** | 66.4 |
| | 3DHP | 89.0 | 89.7 | 85.9 | 109.2 | **70.6** |
| Same-Dataset Error Reduction ↓ | | 0.9 | 0.8 | 0.1 | 3.2 | 0.8 |
| Cross-data Error Reduction ↓ | | 2.9 | 10.6 | 4.3 | 8.7 | 4.7 |

Table 3.6: Baseline cross-dataset test error and error reduction (Procrustese aligned MPJPE) from the addition of our proposed quaternion loss. Bold indicates the best performing model on each the test sets (rows). Blue color indicates test set which saw greatest error reduction.

| | | PCK3D (accuracy, higher is better) | | | | |
|---|---|---|---|---|---|---|
| | Testing \Training | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Baseline | H36M | **95.7** | 75.7 | 52.3 | 70.6 | 77.8 |
| | GPA | 78.3 | **96.3** | 58.8 | 76.2 | 84.5 |
| | SURREAL | 76.4 | 84.5 | **97.2** | 73.6 | 81.0 |
| | 3DPW | 83.2 | 78.7 | 54.5 | **82.1** | 81.7 |
| | 3DHP | 76.1 | 70.3 | 44.8 | 68.4 | **84.2** |
| Our Method | H36M | **96.0** | 78.9 | 52.6 | 72.8 | 78.3 |
| | GPA | 81.5 | **96.8** | 59.3 | 76.4 | 84.8 |
| | SURREAL | 80.0 | 84.8 | **97.3** | 76.2 | 81.3 |
| | 3DPW | 83.2 | 80.8 | 54.7 | **84.6** | 81.7 |
| | 3DHP | 76.1 | 73.5 | 45.1 | 70.3 | **84.3** |
| Same-Dataset Accuracy Increase ↑ | | 0.3 | 0.5 | 0.1 | 2.5 | 0.1 |
| Cross-data Accuracy Increase ↑ | | 6.8 | 8.8 | 1.3 | 6.9 | 1.1 |

Table 3.7: Baseline cross-dataset test accuracy and accuracy increases (PCK3D) from the addition of our proposed quaternion loss. Bold indicates the best performing model on each the test set (rows). Blue color indicates test set which saw greatest accuracy increase.

## 3.8 Quaternion and cluster centers

Instead of colorizing each quaternion with cluster index, we directly visualize quaternion with the same color within each dataset in Fig 3.9, and also plot the cluster centers in the azimuth and elevation space.

| | | MPJPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|---|
| | Testing \Training | H36M | GPA | SURREAL | 3DPW | 3DHP |
| Baseline | H36M | **72.5** | 126.0 | 116.6 | 135.5 | 118.0 |
| | GPA | 110.5 | **76.6** | 97.3 | 116.2 | 100.6 |
| | SURREAL | 129.6 | 116.0 | **54.1** | 132.3 | 118.7 |
| | 3DPW | 120.1 | 121.9 | 120.2 | **108.5** | 119.8 |
| | 3DHP | 122.9 | 133.6 | 128.5 | 148.0 | **104.5** |
| Our Method | H36M | **71.9** | 122.2 | 115.4 | 134.4 | 109.9 |
| | GPA | 109.9 | **72.4** | 97.8 | 115.3 | 102.0 |
| | SURREAL | 129.2 | 113.5 | **53.9** | 126.5 | 119.4 |
| | 3DPW | 119.1 | 119.3 | 119.9 | **100.9** | 116.5 |
| | 3DHP | 122.5 | 130.2 | 127.6 | 145.7 | **103.3** |
| Same-Dataset Error Reduction ↓ | | 0.6 | 4.2 | 0.2 | 7.6 | 1.2 |
| Cross-data Error Reduction ↓ | | 2.4 | 12.3 | 1.9 | 10.1 | 9.3 |

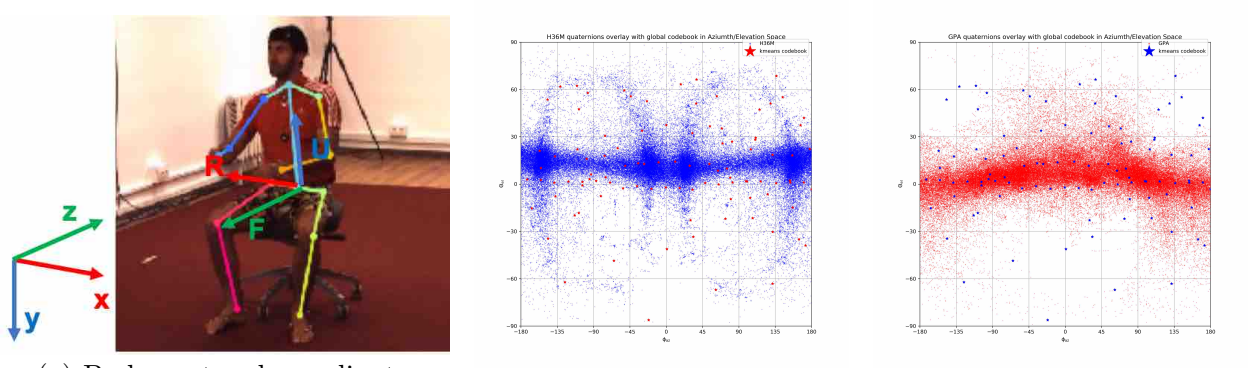
Table 3.8: Retraining the model of Zhou *et al.* [215] using our viewpoint prediction loss also shows significant decrease in prediction error, demonstrating the generality of our finding.

## 3.9 Sampled images from five datasets

**Sampled images from H36M** We sample images from the interesting azimuth/elevation pattern from H36M. We can see the images from Fig 3.10a are facing right while images from Fig 3.10b are facing left. The index in the azimuth/elevation images corresponds with the index on top of images sampled and placed around the center figure.

**Sampled images from GPA/SURREAL** We sample images from SURREAL and GPA with uniform azimuth from left to right, and place some randomness on elevation during sampling. We can see the patterns of sampled images from left to right: facing towards back and rotating to facing right, and facing towards the camera, and then facing back again in Fig 3.11.

**Sampled images from 3DHP** We sample images from 3DHP with uniform azimuth from left to right as shown in Fig 3.12b, uniform elevation from top to down as shown in Fig 3.12c, and from camera center as shown in Fig 3.12a, during sampling we add some randomness on sampled elevation/azimuth around camera centers.




(a) Body-centered coordinate (b) H36M (c) GPA

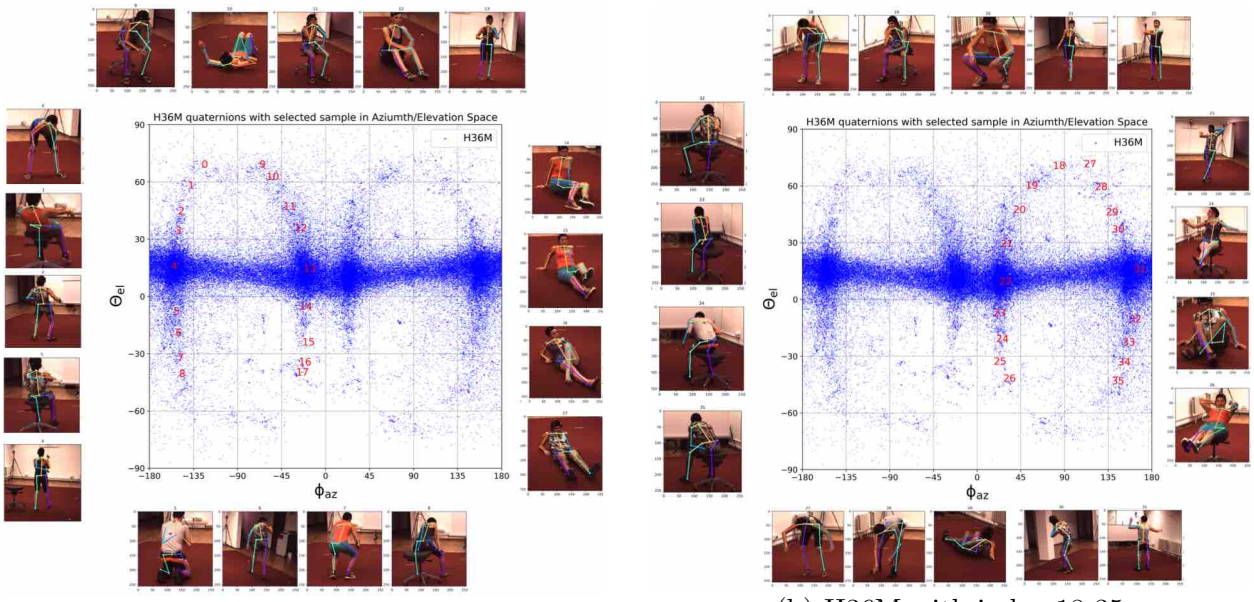
(d) SURREAL (e) 3DPW (f) 3DHP

Figure 3.9: **a**: Illustration of our body-centered coordinate frame (up vector, right vector and front vector) relative to a camera-centered coordinate frame. **b-f**: Camera viewpoint distribution of the 5 datasets overlaid with quaternion cluster centers. Quaternions (rotation between body-centered and camera frame) are sampled from training sets and clustered using k-means.

**Sampled images from 3DPW** We sample images from 3DPW with extreme elevation as shown in Fig 3.13a, and randomly as shown Fig 3.13b.

## 3.10 Qualitative Results

**Qualitative Results trained on four datasets** We visualize the prediction on the 5 datasets with model trained on **GPA**, **SURREAL**, **3DPW**, **3DHP** separately on using our proposed method in Fig 3.14,3.15,3.16,3.17. The 2d joint prediction is overlaid with cropped images while the 3D joint prediction is visualized in our proposed body-centered coordinates.




(a) H36M with index 0-17

(b) H36M with index 18-35

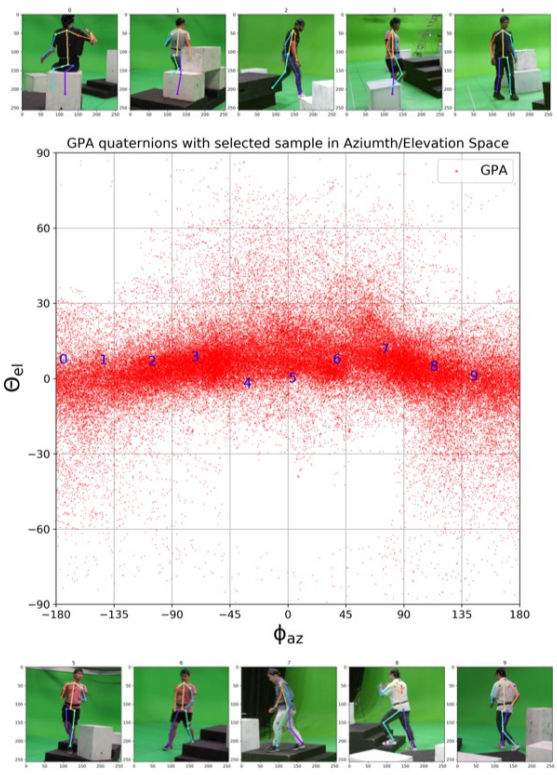
Figure 3.10: H36M and sampled images.

From top to bottom are H36M, GPA, SURREAL, 3DPW and 3DHP datasets. We rank the images from left to right in MPJPE increasing order.

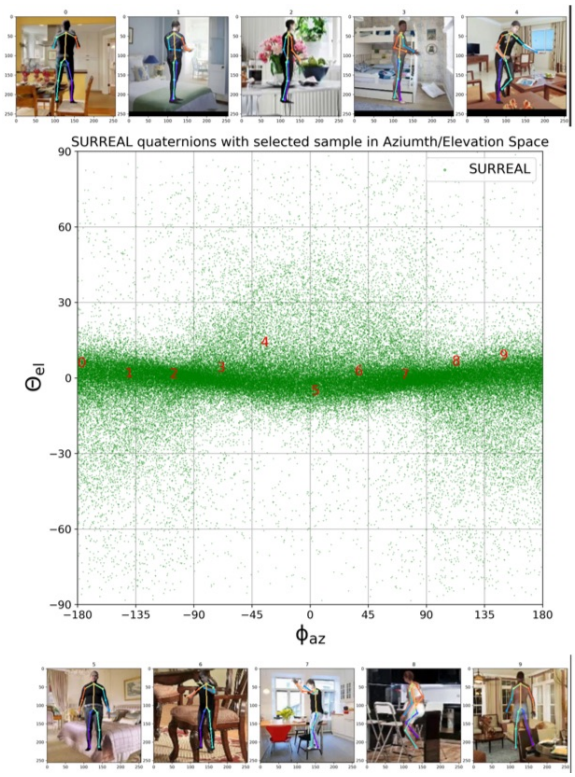
**Qualitative Results tested on the same images** We further visualize the models trained on 5 datasets, and test on images from the dataset H36M in Fig 3.18, GPA in Fig 3.19, SURREAL in Fig 3.20, 3DPW in Fig 3.21 and 3DHP in Fig 3.22. The results from left to right are models trained on H36M, GPA, SURREAL, 3DPW, and 3DHP. The RGB images are overlaid with 2d joint prediction from model trained on each dataset.

## 3.11 Conclusions

In this chapter, we observe strong dataset-specific biases present in the distribution of cameras relative to the human body and propose the use of body-centered coordinate frames. Utilizing the relative rotation between body-centered coordinates and camera-centered coordinates as an additional supervisory signal, we significantly reduce the 3D joint prediction error and



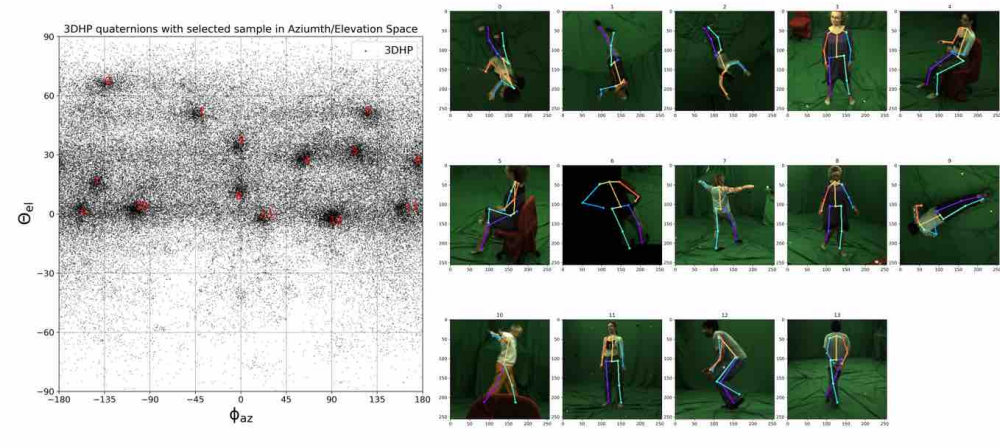

(a) GPA with sampled images.



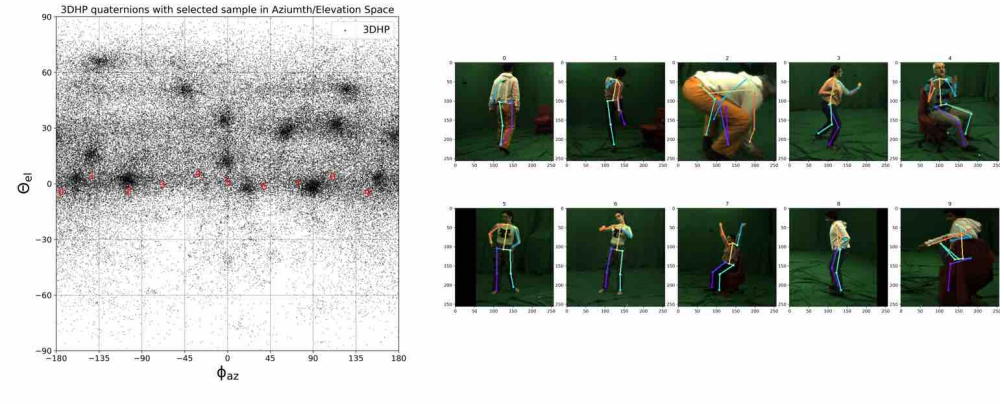

(b) SURREAL with sampled images

Figure 3.11: GPA and SURREAL sampled images.

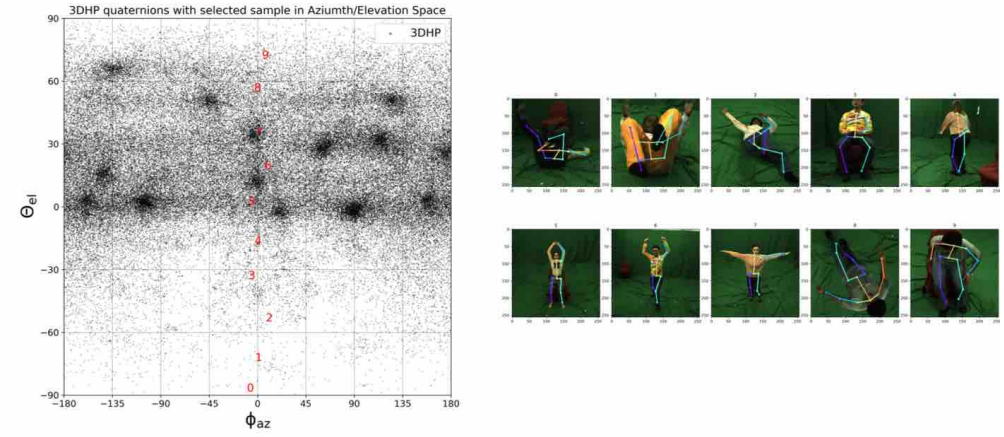
improve generalization in cross-dataset 3D human pose evaluation. Out model also achieves state-of-the-art performance on all same-dataset evaluations. We hope that our cross-dataset analysis is useful for future work and serves as a resource to guide future dataset collection.




(a) 3DHP with images sampled from camera center.

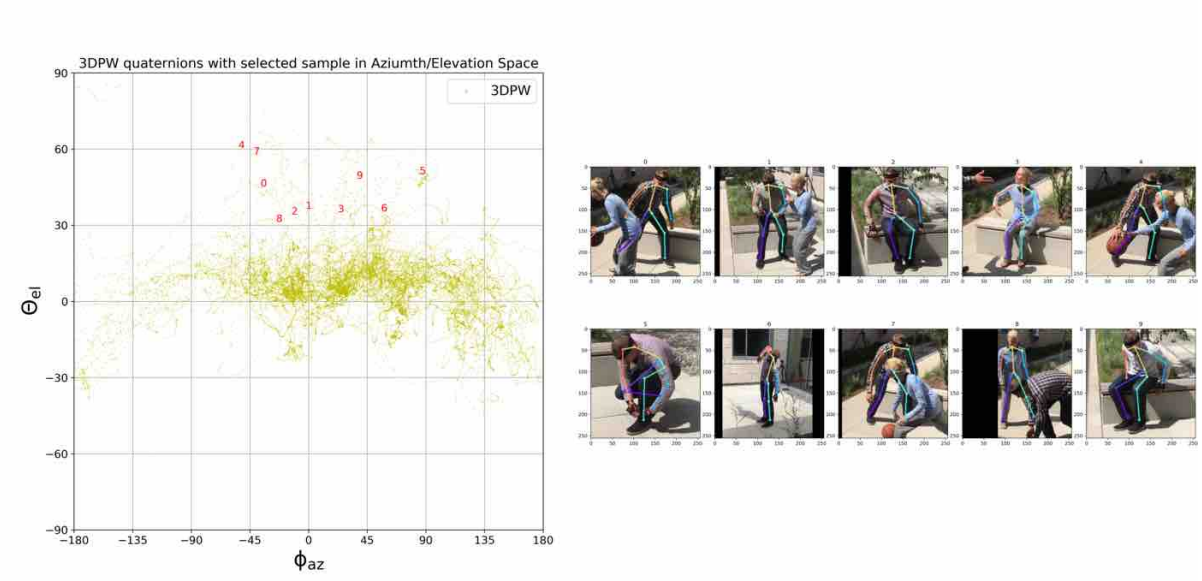



(b) 3DHP with sampled images in uniform azimuth space.

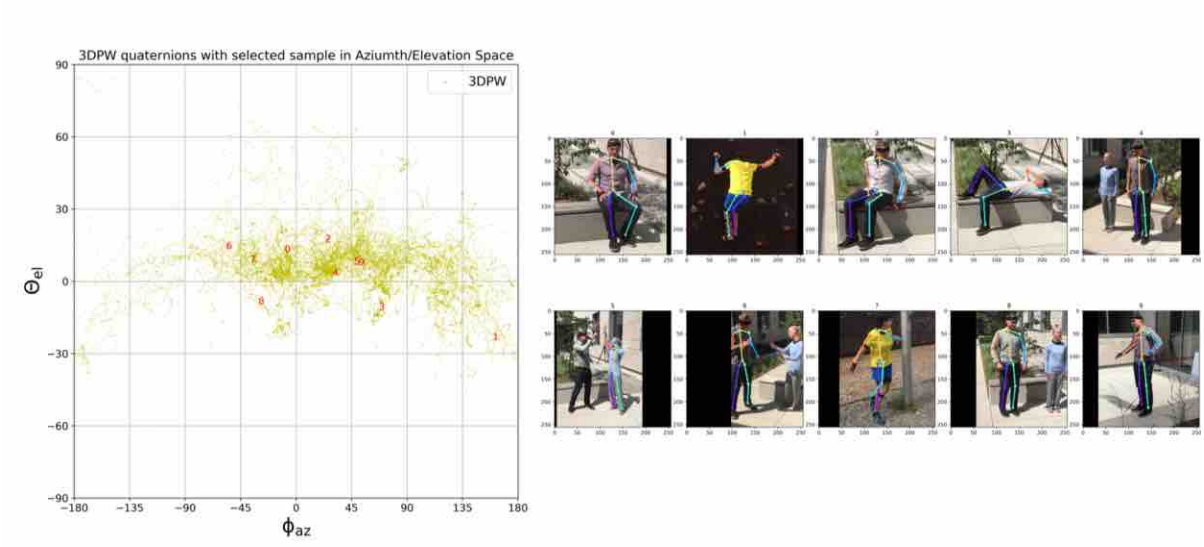



(c) 3DHP with sampled images in uniform elevation space.

Figure 3.12: 3DHP sampled images.




(a) 3DPW with extreme elevation sampled images.




(b) 3DPW with random sampled images.

Figure 3.13: 3DPW sampled images.

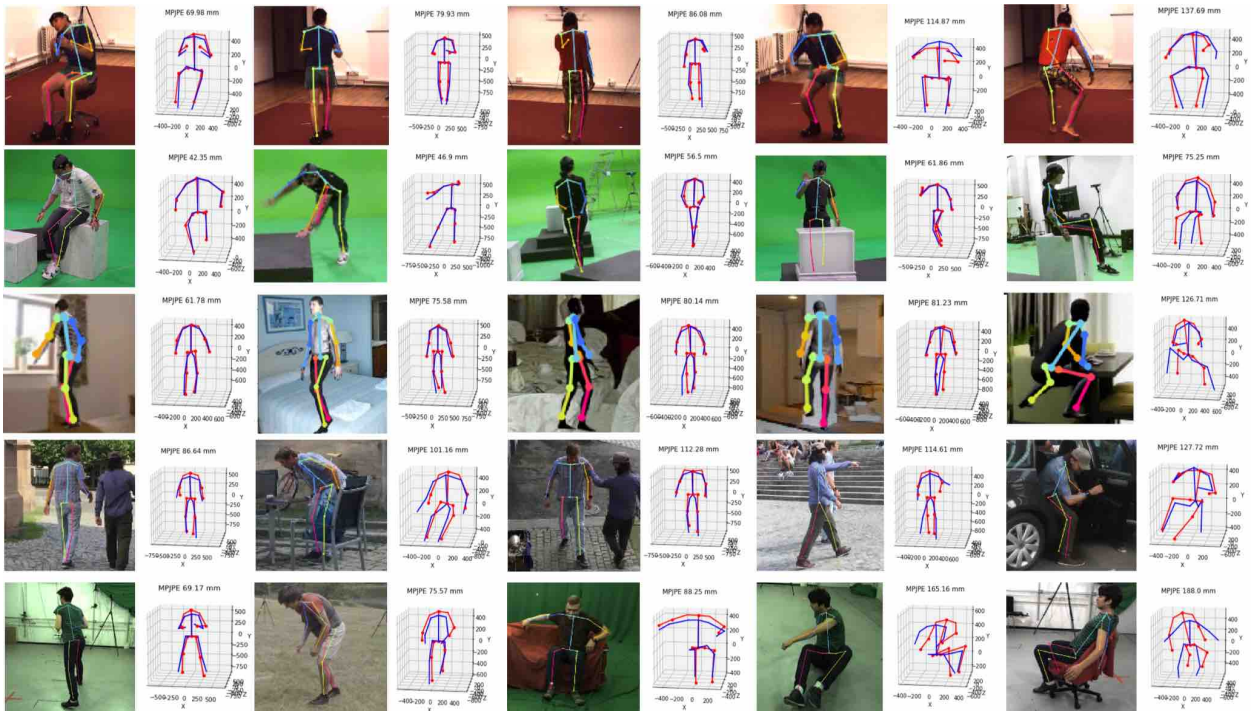


Figure 3.14: Our prediction on 5 diverse dataset with model trained on GPA dataset. The 2d joints are overlaid with the original image, while the 3D prediction (red) is overlaid with 3D ground truth (blue). 3D prediction is **visualized in body-centered coordinate** rotated by the relative rotation between ground truth root-relative coordinate and body-centered coordinate. From top to bottom are H36M, GPA, SURREAL, 3DPW and 3DHP datasets. We rank the images from left to right in MPJPE increasing order.

Figure 3.15: Our prediction on 5 diverse datasets with model trained on SURREAL dataset.

Figure 3.16: Our prediction on 5 diverse datasets with model trained on 3DPW dataset.

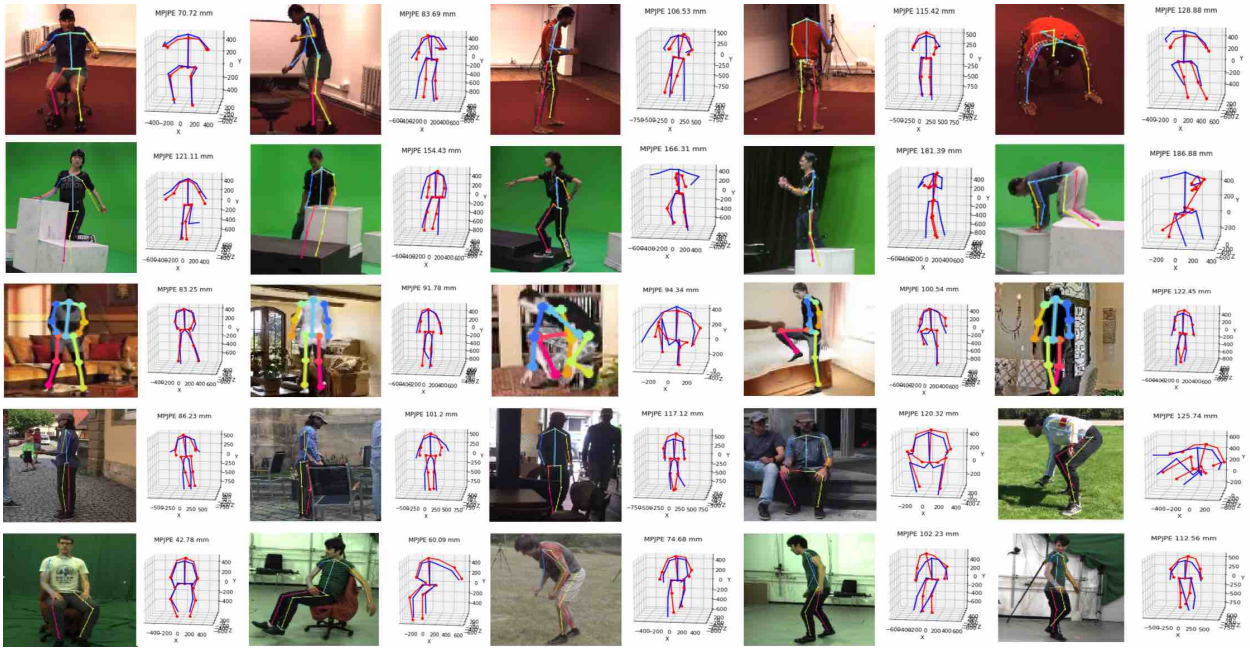


Figure 3.17: Our prediction on 5 diverse datasets with model trained on 3DHP dataset.

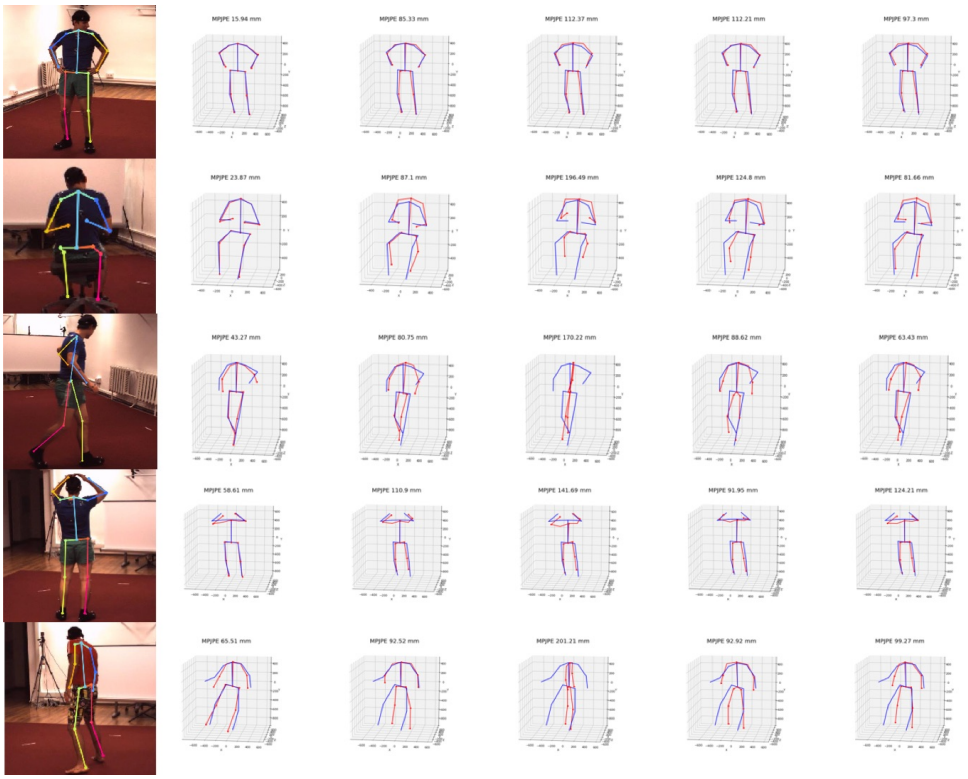


Figure 3.18: Model trained on 5 models tested on the same images from H36M, from left to right (model trained on H36M, GPA, SURREAL, 3DPW, 3DHP).

Figure 3.19: Model trained on 5 models tested on the same images from GPA, from left to right (model trained on H36M, GPA, SURREAL, 3DPW, 3DHP).

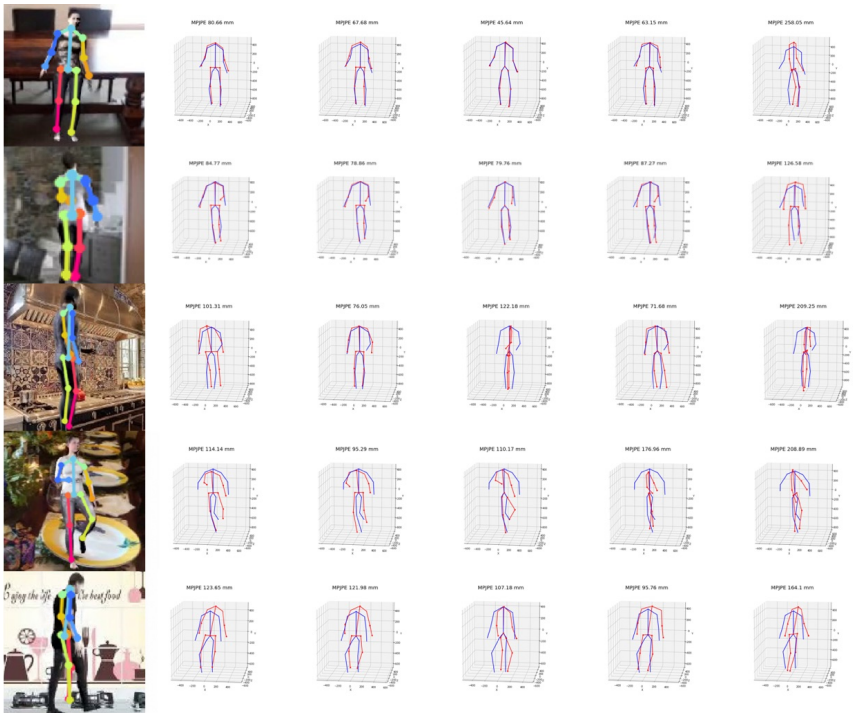


Figure 3.20: Model trained on 5 models tested on the same images from SURREAL, from left to right (model trained on H36M, GPA, SURREAL, 3DPW, 3DHP).

Figure 3.21: Model trained on 5 models tested on the same images from 3DPW, from left to right (model trained on H36M, GPA, SURREAL, 3DPW, 3DHP).

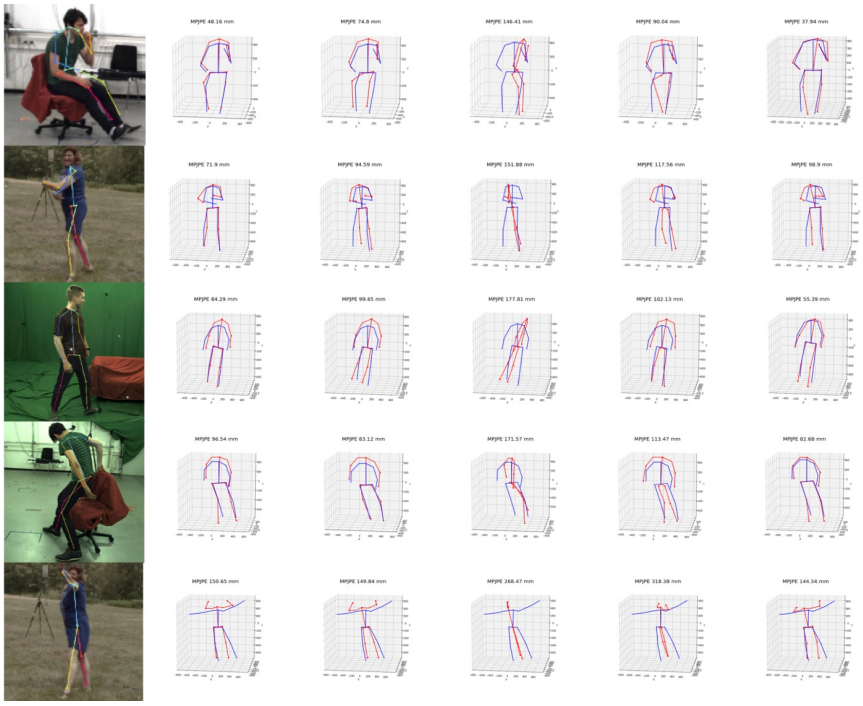


Figure 3.22: Model trained on 5 models tested on the same images from 3DHP, from left to right (model trained on H36M, GPA, SURREAL, 3DPW, 3DHP).

# Chapter 4

# Geometric Pose Affordance: Monocular 3D Human Pose Estimation with Scene Constraints

## 4.1 Introduction

Accurate estimation of human pose in 3D from image data would enable a wide range of interesting applications in emerging fields such as virtual and augmented reality, humanoid robotics, workplace safety, and monitoring mobility and fall prevention in aging populations. Interestingly, many such applications are set in relatively controlled environments (e.g., the home) where large parts of the scene geometry are relatively static (e.g., walls, doors, heavy furniture). We are interested in the following question, *"Can strong knowledge of scene geometry improve our estimates of human pose from images?"*.

Consider the images in Fig. 4.1 a. Intuitively, if we know the 3D locations of surfaces in the scene, this should constrain our estimates of pose. Hands and feet should not interpenetrate

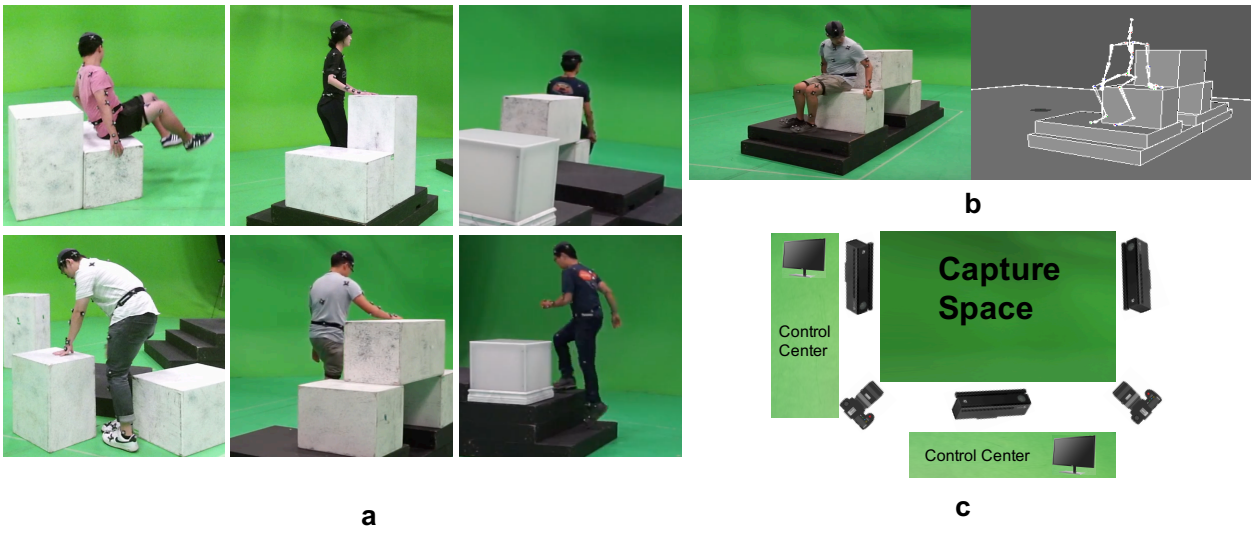



Figure 4.1: **a**: Samples from our data set featuring scene constrained poses: stepping on the stairs, sitting on the tables and touching boxes. **b**: Sample frame of a human interacting with scene geometry, and visualization of the corresponding 3D scene mesh with captured human pose. **c**: Motion capture setup. We simultaneously captured 3 RGBD and 2 RGB video streams and ground-truth 3D pose from a VICON marker-based mocap system. Cameras are calibrated with respect to a 3D mesh model of scene geometry.

scene surfaces, and if we see someone sitting on a surface of known height we should have a good estimate of where their hips are even if large parts of the body are occluded. This general notion of scene affordance [1] has been explored as a tool for understanding functional and geometric properties of a scene [39, 31, 178, 85]. However, the focus of such work has largely been on using estimated human pose to infer scene geometry and function.

Surprisingly, there has been little demonstration of how scene knowledge can constrain pose estimation. Traditional 3D pose estimation models have explored kinematic and dynamic constraints which are scene agnostic and have been tested on datasets of people freely performing actions in large empty spaces. *We posit one reason that scene constraints have not been utilized is lack of large-scale datasets of annotated 3D pose in rich environments.* Methods have been developed on datasets like Human3.6M [52] and MPI-INF-3DHP [101], which lack diverse scene geometry (at most one chair or sofa) and are generally free from

[1] "The meaning or value of a thing consists of what it affords." -JJ Gibson (1979)

scene occlusion. Recent efforts have allowed for more precise 3D pose capture for in-the-wild environments [173] but lack ground-truth scene geometry, or provide scene geometry but lack extensive ground-truth pose estimates [44].

Instead of tackling human pose estimation in isolation, we argue that systems should take into account available information about constraints imposed by complex environments. A complete solution must ultimately tackle two problems: (i) estimating the geometry and free space of the environment (even when much of that free space is occluded from view), (ii) integrating this information into pose estimation process. Tools for building 3D models of static environments are well developed and estimation of novel scene geometry from single-view imagery has also shown rapid progress. Thus, we focus on the second aspect under the assumption that high-quality geometric information is available as an input to the pose estimation pipeline.

The question of how to represent geometry and incorporate the constraints it imposes with current learning-based approaches to modeling human pose is an open problem. There are several candidates for representing scene geometry: voxel representations of occupancy [118] are straightforward but demand significant memory and computation to achieve reasonable resolution; Point cloud [11] representations provide more compact representations of surfaces by sampling but lack topological information about which locations in a scene constitute free space. Instead, we propose to utilize *multi-layer depth maps* [144] which provide a compact and nearly complete representation of scene geometry that can be readily queried to verify pose-scene consistency.

We develop and evaluate several approaches to utilize information contained in the multi-layer depth map representation. Since multi-layer depth is a view-centered representation of geometry, it can be readily incorporated as an additional input feature channel. We leverage estimates of 2D pose either as a heatmap or regressed coordinate and query the multi-layer depth map directly to extract features encoding local constraints on the z-coordinates of joints

that can be used to predict geometry-aware 3D joint locations. Additionally, we introduce a differentiable loss that encourages a model trained with such features to respect hard constraints imposed by scene geometry. We perform an extensive evaluation of our multi-layer depth map models on a range of scenes of varying complexity and occlusion. We provide both qualitative and quantitative evaluation on real data demonstrating that these mechanisms for incorporating geometric constraints improves upon scene-agnostic state-of-the-art methods for 3D pose estimation.

To summarize our main contributions: 1. We collect and curate a unique, large-scale 3D human pose estimation dataset with rich ground-truth scene geometry and a wide variety of pose-scene interactions (see e.g. Fig. 4.1) 2. We propose a novel representation of scene geometry constraints: multi-layer depth map, and explore multiple ways to incorporate geometric constraints into contemporary learning-based methods for predicting 3D human pose. 3. We experimentally demonstrate the effectiveness of integrating geometric constraints relative to two state-of-the-art scene-agnostic pose estimation methods.

| Dataset | Frames | Scenes | Characteristics |
|---|---|---|---|
| HumanEva (2010) | 80k | ground plane | marker-based pose and video |
| Human36M (2014) | 3.6M | chairs | marker-based, human body scans |
| MPI-INF-3DHP (2017) | 3k | chairs, sofa | marker-less, indoor and outdoor backgrounds |
| TotalCapture (2017) | 1.9M | ground plane | marker-based pose, IMU and video |
| Surreal (2017) | 6M | ground plane | synthetic renderings of CMU Mocap data |
| Ski-Pose (2018) | 10k | ski slope | marker-less using multi-view 2D annotation |
| 3DPW (2018) | 51k | in the wild | IMU-based capture with mobile camera |
| **GPA (2019)** | 0.7M | boxes, chairs, stairs | scene interaction, geometry ground-truth |

Table 4.1: Comparison of existing datasets commonly used for training and evaluating 3D human pose estimation methods. Previous datasets have primarily focused on capturing a diverse range human motions, actions, abd subjects using optical markers and/or IMUs to establish ground-truth pose. Our dataset focuses on interactions between humans and static scene geometry and includes both ground-truth 3D pose and a complete description of the scene geometry.

## 4.2 Related Work

**Motion capture for ground-truth 3D pose** The work of [145] introduced one of the first large-scale 3D human pose estimation datasets with synchronized images and ground-truth 3D keypoint locations. [52] scaled their dataset up to 3.6 million images covering a range of subjects and actions along with depth images and 3D body scans of the human subjects. To overcome the limitations of marker-based data collection such as constrained clothing and capture environment, several marker-less approaches have also been used. [59] utilize an indoor "panoptic studio" to capture poses from 10 calibrated RGBD cameras. [101] utilized multi-view marker-less capture to collect pose data for subjects wearing a variety of clothing against both indoor and outdoor backgrounds. [134] utilized calibrated PTZ cameras and human annotators to triangulate joint locations skiers over a large area of a ski-slope. [217] also explores motion capture both indoor and outdoor using a Drone. Synchronized inertial measurement sensor (IMU) data can be used to further enhance marker-less capture. [164] develop an approach to fusing inertial measurement sensors with multi-view recording in a studio environment. [101] use an IMU-based system along with a single synchronized mobile camera video stream to capture 3D human pose "in the wild".

These data collection efforts have largely focused on covering a diverse range of poses and actions, but actions take place in simple environments (i.e., an empty room) which minimize occlusion and impose very few geometric affordance constraints on human pose. Recent "in the wild" markerless capture data such as [101] encompass much richer environments, but the scene geometry is unknown. In contrast, our dataset provides gold-standard, marker-based 3D pose of subjects in richer environments with ground-truth scene geometry, offering a controlled test-bed for research in 3D human pose estimation with rich geometric affordance. [155] collects a dataset for grasping, with the markers placed both on hands and on bodies to capture whole-body pose during grasping and object manipulation. This is complementary to our dataset as it provides object geometry and grasping contacts while our dataset

samples whole-body affordance. We provide a summary comparison of recent 3D human pose estimation datasets in Table 4.1.

**Modeling scene affordances** The term "affordance" was coined by J Gibson [33] to capture the notion that the meaning and relevance of many objects in the environment are largely defined in relation to the ways in which an individual can functionally interact with them. For computer vision, this suggests scenarios in which the natural labels for some types of visual content may not be semantic categories or geometric data but rather functional labels, i.e., which human interactions they afford. [39] present a human-centric paradigm for scene understanding by modeling physical human-scene interactions. [31] rely on pose estimation methods to extract functional and geometric constraints about the scene and use those constraints to improve estimates of 3D scene geometry. [178] collects a large-scale dataset of images from sitcoms which contains multiple images of the same scene with and without humans present. Leveraging state-of-the-art pose estimation and generative model to infer what kind of poses each sitcom scene affords. [85] build a fully automatic 3D pose synthesizer to predict semantically plausible and physically feasible human poses within a given scene. [106] applies an energy-based model on synthetic videos to improve both scene and human motion mapping. [9] construct a synthetic dataset utilizing a game engine. They first sample multiple human motion goals based on a single scene image and 2D pose histories, plan 3D human paths towards each goal, and finally predict 3D human pose sequences following each path. Rather than labeling image content based on observed poses, our approach is focused on estimating scene affordance directly from physical principles and geometric data, and then subsequently leveraging affordance to constrain estimates of human pose and interactions with the scene.

Our work is also closely related to earlier work on scene context for object detection. [48, 47] used estimates of ground-plane geometry to reason about location and scales of objects in

an image. More recent work such as [176, 24, 99] use more extensive 3D models of scenes as context to improve object detection performance. Geometric context for human pose estimation differs from generic object detection in that humans are highly articulated. This makes incorporating such constraints more complicated as the resulting predictions should simultaneously satisfy both scene-geometric and kinematic constraints.

**Constraints in 3D human pose estimation** Estimating 3D human pose from monocular image or video is an ill-posed problem that can benefit from prior constraints. Recent examples include [29] who model kinematics, symmetry and motor control using an RNN when predicting 3D human joints directly from 2D key points. [196] propose an adversarial network as an anthropometric regularizer. [175, 220] construct a graphical model encoding priors to fit 3D pose reconstruction. [139, 13] first build a large set of valid 3D human poses and treat estimation as a matching or classification problem. [1, 133] explore joint constraints in 3D and geometric consistency from multi-view images. [215] improve joint estimation by adding bone-length ratio constraints.

To our knowledge, there is relatively little work on utilizing scene constraints for 3D human pose. [203] utilize an energy-based optimization model for pose refinement which penalizes ankle joint estimates that are far above or below an estimated ground-plane. The recent work of [44] introduces scene geometry penetration and contact constraints in an energy-based framework for fitting parameters of a kinematic body model to estimate pose. In our work, we explore a complementary approach which uses CNN-based regression models that are trained to directly predict valid pose estimates given image and scene geometry as input.

## 4.3 Geometric Pose Affordance Dataset (GPA)

To collect a rich dataset for studying interaction of scene geometry and human pose, we designed a set of action scripts performed by 13 subjects, each of which takes place in one of 6 scene arrangements. In this section, we describe the dataset components and the collection process.

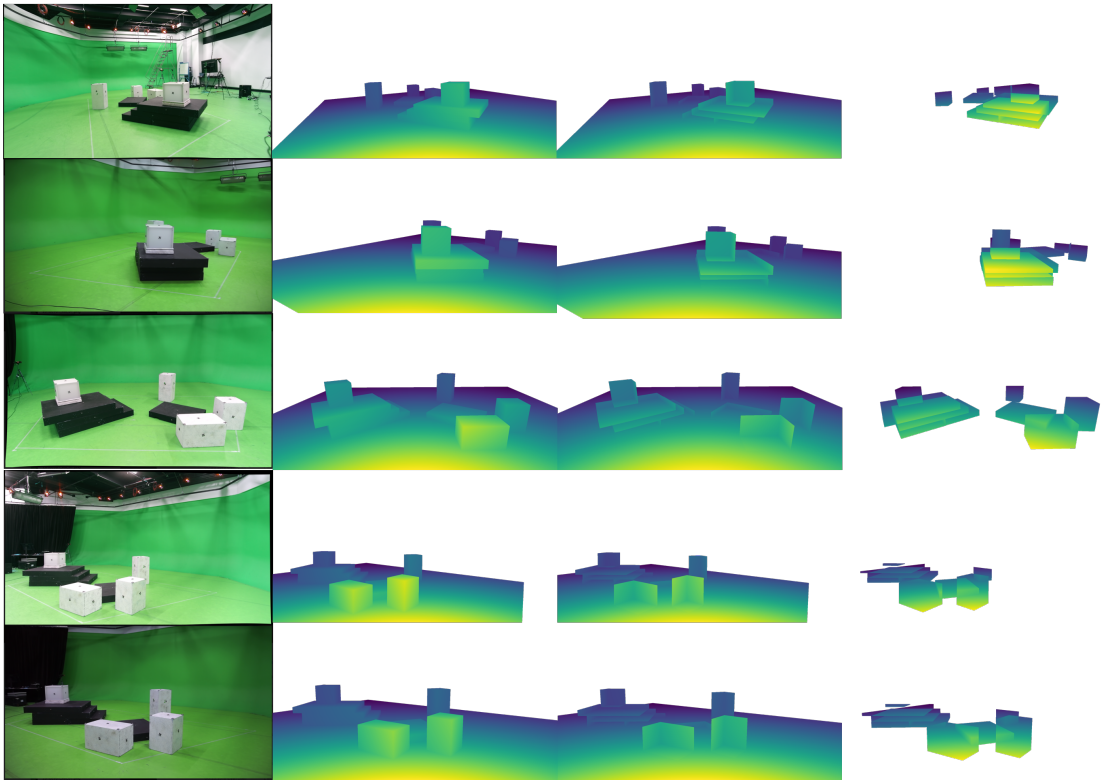


Figure 4.2: The 5 camera views from the same scene with the first 3 layers of corresponding multi-layer depth map (for visualization clarity, we plot inverse depth). 2nd column corresponds to a traditional depth map, recording the depth of the first visible surface in the scene from the camera viewpoint of 1st column. 3rd column is when the multi-hit ray leaves the first layer of objects (e.g. the backside of the boxes). 4th column is when the multi-hit ray hits another object.

### 4.3.1 Human Poses and Subjects

We designed three action scripts that place emphasis on semantic actions, mechanical dynamics of skeletons, and pose-scene interactions. We refer to them as *Action, Motion, and Interaction Sets* respectively. The semantic actions of *Action Set* are constructed from a subset of Human3.6M [52], namely, *Direction, Discussion, Writing, Greeting, Phoning, Photo, Posing* and *Walk Dog* to provide a connection for comparisons between our dataset and the de facto standard benchmark. *Motion Set* includes poses with more dynamic range of motion, such as running, side-to-side jumping, rotating, jumping over obstacles, and improvised poses from subjects. *Interaction Set* mainly consists of close interactions between body parts and surfaces in the scene to support modeling geometric affordance in 3D. There are three main poses in this group: *Sitting, Touching, Standing on*, corresponding to typical affordance relations *Sittable, Walkable, Reachable* [31, 39]. The 13 subjects included 9 males and 4 female with roughly the same age and medium variations in heights approximately from 155cm to 190cm, giving comparable subject diversity to Human3.6M.

### 4.3.2 Image Recording and Motion Capture

This motion capture studio layout is also illustrated in Fig. 4.1 c. We utilized two types of camera, RGBD and RGB, placed at 5 distinct locations in the capture studio. All 5 cameras have a steady 30fps frame rate but their time stamps are only partially synchronized, requiring additional post-processing described below. The color sensors of the 5 cameras have the same 1920x1080 resolution and the depth sensor of the Kinect v2 cameras has a resolution at 640x480. The motion capture system was a standard VICON system with 28 pre-calibrated cameras covering the capture space which are used to estimate the 3D coordinates of IR-reflective tracking markers attached to the surface of subjects and objects.

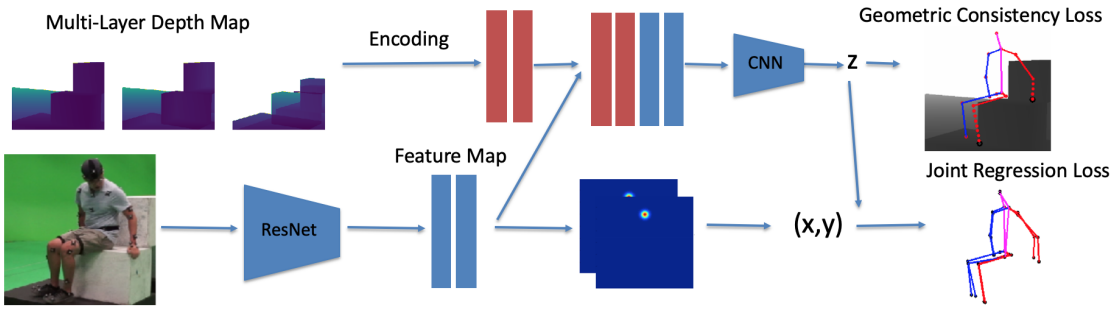



Figure 4.3: Overview of model architecture: we use ResNet-50 as our backbone to extract features from a human centered cropped image. The feature map is used to predict 2D joint location heatmaps and is also concatenated with encoded multi-layer depth map. The concatenated feature is used to regress the depth (z-coordinate) of each joint. The model is trained with a loss on joint location (joint regression loss) and scene affordance (geometric consistency loss). The 2d joint heatmaps are decoded to x,y joint locations using an argmax. The geometric consistency loss is described in more detail in Fig 4.6 (a) and Section 4.2.

### 4.3.3 Scene Layouts

Unlike previous efforts that focus primarily on human poses without other objects present (e.g. [52, 101]), we introduced a variety of scene geometries with arrangements of 9 cuboid boxes in the scene. The RGB images captured from 5 distinct viewpoints exhibit substantially more occlusion of subjects than existing datasets (as illustrated in Fig 4.1 and Fig 4.2) and constrain the set of possible poses. We captured 1 or 2 subjects interacting with each scene and configured a total of 6 distinct scene geometries.

To record static scene geometry, we measured physical dimension of all the objects (cuboids) as well as scanning the scene with a mobile Kinect sensor. We utilized additional motion-capture markers attached to the corners and center face of each object surface so that we could easily align geometric models of the cuboids with the global coordinate system of the motion capture system. We also use the location of these markers, when visible in the RGB capture cameras, in order to estimate extrinsic camera parameters in the same global coordinate system. This allows us to quickly create geometric models of the scene which are well aligned to all calibrated camera views and the motion capture data.

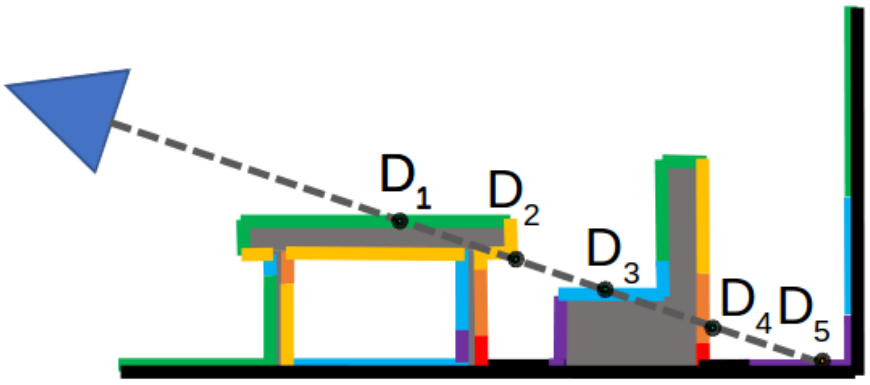



Figure 4.4: Illustration of multi-layer depth map. For each image pixel we record the depth of all surface intersections along the view ray (e.g., $D_1, D_2, D_3, D_4, D_5$ ).

### 4.3.4 Scene Geometry Representation

Mesh models of each scene were initially constructed in global coordinates using modeling software (Maya) with assistance from physical measurements and reflective markers attached to scene objects. To compactly represent the scene geometry from the perspective of a given camera viewpoint, we utilize a multi-layer depth map. *Multi-layer depth maps* are defined as a map of camera ray entry and exit depths for all surfaces in a scene from a given camera viewpoint (illustrated in Fig 4.4). Unlike standard depth-maps which only encode the geometry of visible surfaces in a scene (sometimes referred to as 2.5D), multi-layer depth provides a nearly[2] complete, viewer-centered description of scene geometry which includes occluded surfaces.

The multi-layer depth representation can be computed from the scene mesh model by performing multi-hit ray tracing from a specified camera viewpoint. Specifically, the multi-hit ray tracing sends a ray from the camera center towards a point on the image plane that corresponds to the pixel at $(x, y)$ and outputs distance values $\{t_1, t_2, t_3, ..., t_k\}$ where $k$ is the total number of polygon intersections along the ray. Given a unit ray direction $\mathbf{r}$ and camera viewing direction $\mathbf{v}$, the depth value at layer $i$ is $D_i(x, y) = t_i \mathbf{r} \cdot \mathbf{v}$ if $i <= k$ and $D_i(x, y) = \varnothing$

[2]Surfaces tangent to a camera view ray are not represented

if $i > k$. In our scenes, the number of multi-layer depth maps is set to 15 which suffices to cover all scene surfaces in our dataset. We visualize 5 camera viewpoints together with first 3 layers of depth map in the same scene in Fig 4.2.

### 4.3.5 Data Processing Pipeline

The whole data processing pipeline includes validating motion capture pose estimates, camera calibration, joint temporal alignment of all data sources, and camera calibration. Unlike previous marker-based mocap datasets which have few occlusions, many markers attached to the human body are occluded in the scene during our capture sessions due to scene geometry. We spent 4 months on pre-processing with help of 6 annotators in total. There are three stages of generating ground truth joints from recorded VICON sessions: **(a)** recognizing and labeling recorded markers in each frame to 53 candidate labels which included three passes to minimize errors; **(b)** applying selective temporal interpolation for missing markers based on annotators' judgement. **(c)** removing clips with too few tracked markers. After the annotation pipeline, we compiled recordings and annotations into 61 sessions captured at 120fps by the VICON software. To temporally align these compiled ground-truth pose streams to image capture streams, we first had annotators to manually correspond 10-20 pose frames to image frames. Then we estimated temporal scaling and offset parameters using RANSAC [30], and regress all timestamps to a single global timeline.

The RGB camera calibration was performed by having annotators mark corresponding image coordinates of visible markers (whose global 3D coordinates are known) and estimating extrinsic camera parameters from those correspondences. We performed visual inspection on all clips to check that the estimated camera parameters yield correct projections of 3D markers to their corresponding locations in the image. With estimated camera distortion parameters, we correct the radial and lens distortions of the image so that they can be

treated as projections from ideal pinhole cameras in later steps. Finally, the scene geometry model was rendered into multi-layer depth maps for each calibrated camera viewpoint. We performed visual inspection to verify that the depth edges in renderings were precisely aligned with object boundaries in the RGB images.

After temporal and geometric calibration, we generated a unified dataset by using an adaptive sampling approach to select non-redundant frames. We consider frames with sufficiently different poses from adjacent ones as "interesting". Here, the measure of difference between two skeleton poses is defined as the 75th percentile of L2 distances between corresponding joints (34 pairs per skeleton pair). This allows us to retain frames where only a few body parts moved significantly while being robust to inter-frame differences due to noise or missing markers. With the measure of difference defined, we select the frames by choosing the change threshold as the 55th percentile, retaining 45% of total frames from the original sequences. This final dataset, which we call Geometric Pose Affordance (GPA) contains 304.9k images, each with corresponding ground-truth 3D pose and scene geometry[3].

### 4.3.6 Dataset Visualization and Statistics

A video demonstrating the output of this pipeline is available online [4]. The video shows the full frame and a crop with ground-truth joints/markers overlayed, for 10 sample clips from the 'Action' and 'Motion' sets. The video also indicates various diagnostic metadata including the video and mocap time stamps, joint velocities, and number of valid markers (there are 53 markers and 34 joints for VICON system). Since we have an accurate model of the scene geometry, we can also automatically determine which joints and markers are occluded from the camera viewpoint.

Fig. 4.5 summarizes statistics on the number of occluded joints as well as the distribution of

[3]The dataset is available online: `https://wangzheallen.github.io/GPA`
[4]Video Link: `https://youtu.be/ZRnCBySt2fk`

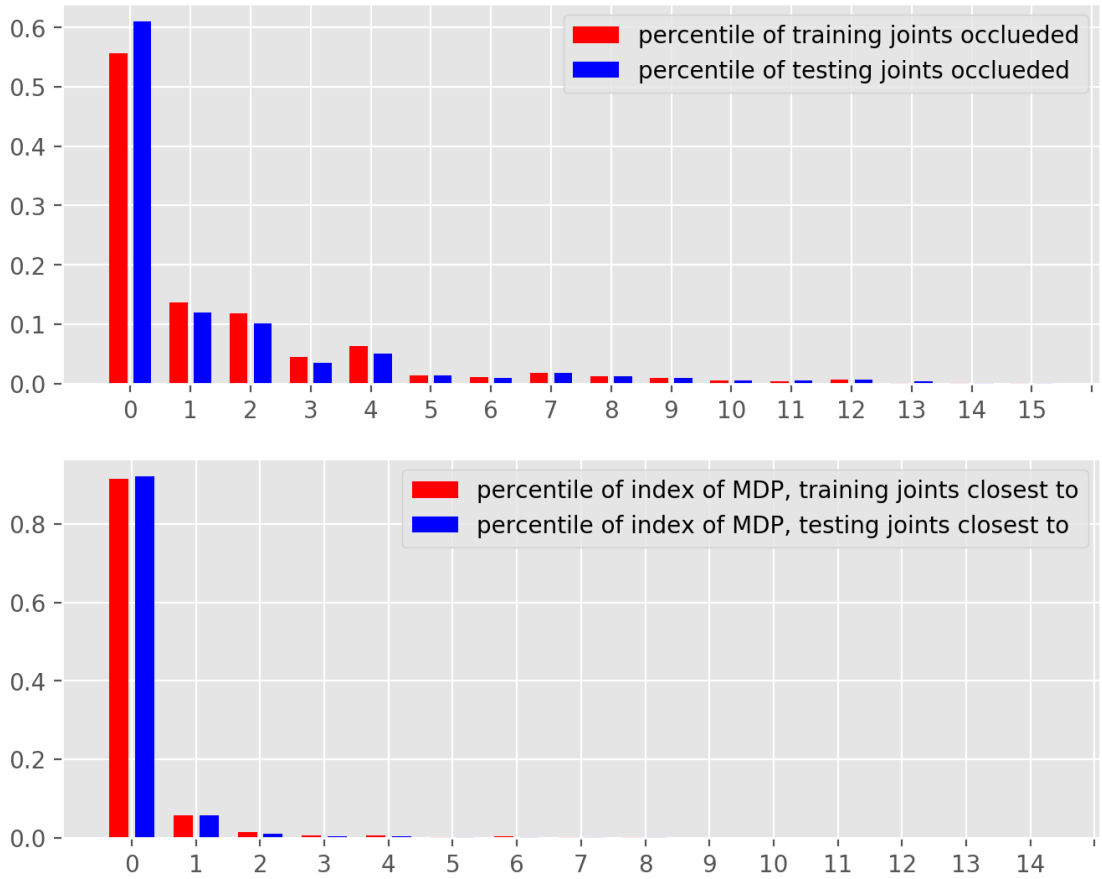



Figure 4.5: Top: Distribution of the number of joints occluded in training and testing frames. Bottom: Distribution of the index of the depth layer closest to each pose. High index layers, which often correspond to hidden surfaces such as the bottom side of platforms, seldom constrain pose.

which multi-depth layer is closest to a joint. While the complete scene geometry requires 15 depth layers, as the figure shows only the first 5 layers are involved in 90% of the interaction between body joints and scene geometry. The remaining layers often represent surfaces which are inaccessible (e.g., bottoms of cuboids).

## 4.4 Geometry-aware Pose Estimation

We now introduce two approaches for incorporating geometric affordance in CNN-based pose regression, building on the baseline architecture of [215]. Given an image $I$ of a human

subject, we aim to estimate the 3D human pose represented by a set of 3D joint coordinates of the human skeleton, $P \in \mathbb{R}^{J\times 3}$ where $J$ is the number of joints. We follow the convention of representing each 3D coordinate in the local camera coordinate system associated with $I$. The first two coordinates are given by image pixel coordinates and the third coordinate is the joint depth in metric coordinates (e.g., millimeters) relative to the depth of a specified root joint. We use $P_{XY}$ and $P_Z$ respectively as short-hand notations for the components of $P$.

### 4.4.1 Pose Estimation Baseline Model

We adopt one popular ResNet-based network described by [190] as our 2D pose estimation module. The network output is a set of low-resolution heat-maps $\hat{S} \in \mathbb{R}^{64\times 64\times J}$, where each map $\hat{S}[:,:,j]$ can be interpreted as a probability distribution over the j-th joint location. At test time, the 2D prediction $\hat{P}_{XY}$ is given by the most probable ($\arg\max$) locations in $S$. This heat-map representation is convenient as it can be easily combined (e.g., concatenated) with the other spatial feature maps. To train this module, we utilize squared error loss

$$\ell_{2D}(\hat{S}|P) = \|\hat{S} - G(P_{XY})\|^2 \tag{4.1}$$

where $G(\cdot)$ is a target distribution created from ground-truth $P$ by placing a Gaussian with $\sigma = 3$ at each joint location.

To predict the depth of each joint, we follow the approach of [215], which combines the 2D joint heatmap and the intermediate feature representations in the 2D pose module as input to a joint depth regression module (denoted **ResNet** in the experiments). These shared visual features provide additional cues for recovering full 3D pose. We train with a smooth $\ell_1$ loss [131] given by:

$$\ell_{1s}(\hat{P}|P) = \begin{cases} \frac{1}{2}\|\hat{P}_Z - P_Z\|^2 & \|\hat{P}_Z - P_Z\| \leq 1 \\ \|\hat{P}_Z - P_Z\| - \frac{1}{2} & \text{o.w.} \end{cases} \tag{4.2}$$

**Alternate baseline:** We also evaluated two alternative baseline architectures. First, we used the model of [98] which detects 2D joint locations and then trains a multi-layer perceptron to regress the 3D coordinates $P$ from the vector of 2D coordinates $P_{XY}$. We denote this simple lifting model as **SIM** in the experiments. To detect the 2D locations we utilized the ResNet model of [190] and also considered an upper-bound based on lifting the ground-truth 2D joint locations to 3D. Second, we trained the **PoseNet** model proposed in [107] which uses integral regression [151] in order to regress pose from the heat map directly.

### 4.4.2 Geometric Consistency Loss and Encoding

To inject knowledge of scene geometry we consider two approaches, *geometric consistency loss* which incorporates scene geometry during training, and *geometric encoding* which assumes scene geometry is also available as an input feature at test time.

**Geometric consistency loss:** We design a geometric consistency loss (GCL) that specifically penalizes errors in pose estimation which violate scene geometry constraints. The intuition is illustrated in Fig. 4.6. For a joint at 2D location $(x, y)$, the estimated depth $z$ should lie within one of a disjoint set of intervals defined by the multi-depth values at that location.

To penalize a joint prediction $P^j = (x, y, z)$ that falls inside a region bounded by front-back surfaces with depths $D_i(x, y)$ and $D_{i+1}(x, y)$ we define a loss that increases linearly with the

penetration distance inside the surface:

$$\ell_{G(i)}(\hat{P}^j|D) = min(max(0, \hat{P}^j_Z - D_i(\hat{P}^j_{XY})), \quad max(0, D_{i+1}(\hat{P}^j_{XY}) - \hat{P}^j_Z)) \tag{4.3}$$

Our complete geometric consistency loss penalizes predictions which place any joint inside the occupied scene geometry

$$\ell_G(\hat{P}|D) = \sum_j \max_{i \in \{0,2,4,\ldots\}} \ell_{G(i)}(\hat{P}^j|D) \tag{4.4}$$

Assuming $\{D_i\}$ is piece-wise smooth, this loss is differentiable almost everywhere and hence amenable to optimization with stochastic gradient descent. The gradient of the loss "pushes" joint location predictions for a given example to the surface of occupied volumes in the scene.

**Encoding local scene geometry:** When scene geometry is available at test time (e.g., fixed cameras pointed at a known scene), it is reasonable to provide the model with an encoding of the scene geometry as input. Our view-centered multi-depth representation of scene geometry can be naturally included as an additional feature channel in a CNN since it is the same dimensions as the input image. We considered two different encodings of multi-layer depth. (1) We crop the multi-layer depth map to the input frame, re-sample to the same resolution as the 2D heatmap using nearest-neighbor interpolation, and offset by the depth of the skeleton root joint. (2) Alternately, we consider a volumetric encoding of the scene geometry by sampling 64 depths centered around the root joint using a range based on the largest residual depth between the root and any other joint seen during training (approx. $+/-1m$). For each $(x, y)$ location and depth, we evaluate the geometric consistency loss $\ell_G$ at that point. This resulting encoding is of size $H \times W \times 64$ and encodes the local volume occupancy around the pose estimate.

For the joint depth regression-based models (**ResNet-***) we simply concatenated the encoded

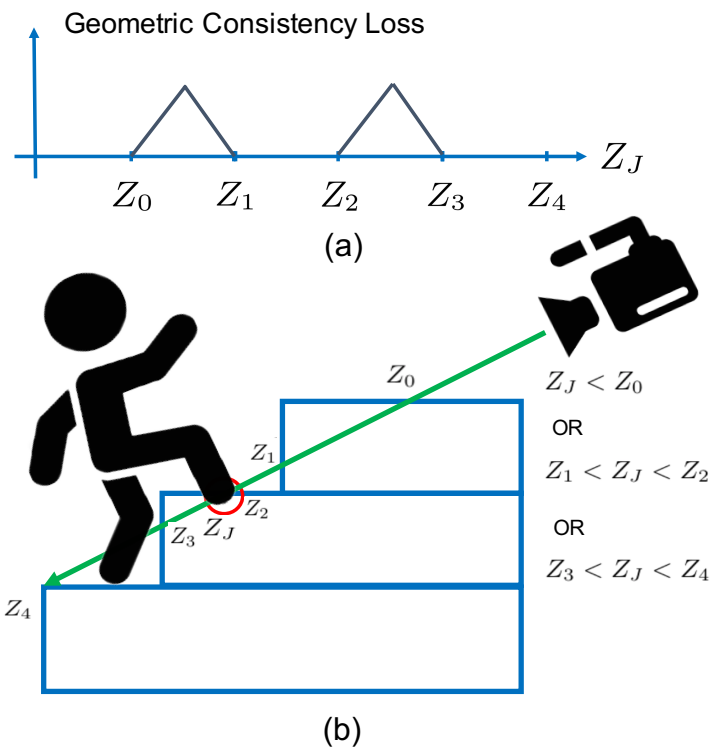



Figure 4.6: (a) is the illustration of the geometry consistency loss as a function of depth along a specific camera ray corresponding to a predicted 2D joint location. In (b) the green line indicates the ray corresponding to the 2D location of the right foot. Our multi-depth encoding of the scene geometry stores the depth to each surface intersection along this ray (i.e., the depth values $Z_0, Z_1, Z_2, Z_3, Z_4$). Valid poses must satisfy the constraint that the joint depth falls in one of the intervals: $Z_J < Z_0$ or $Z_1 < Z_J < Z_2$ or $Z_3 < Z_J < Z_4$. The geometric consistency loss pushes the prediction $Z_J$ towards the closest valid configuration along the ray, $Z_J = Z_2$ .

multi-depth as additional feature channels. For the lifting-based models (**SIM-***), we query the multi-depth values at the predicted 2D joint locations and use the results as additional inputs to the lifting network.

In our experiments we found that the simple and memory efficient multi-layer depth encoding (1) performed the same or better than volumetric encoding with ground-truth root joint offset. However, the volumetric encoding (2) was more robust when there was noise in the root joint depth estimate.

### 4.4.3 Overall Training

Combining the losses in Eq. 4.1, 4.2, and 4.4, the total loss for each training example is

$$\ell(\hat{P}, \hat{S}|P, D) = \ell_{2D}(\hat{S}|P) + \ell_{1s}(\hat{P}|P) + \ell_G(\hat{P}|P, D)$$

We follow [215] and adopt a stage-wise training approach: Stage 1 initializes the 2D pose module using 2D annotated images (i.e., MPII dataset); Stage 2 trains the 3D pose estimation module, jointly optimizing the depth regression module as well as the 2D pose estimation module; Stage 3 of training adds the geometry-aware components (encoding input, geometric consistency loss) to the modules trained in stage 2.

| **Set** | **Number of Images** |
|---|---|
| Full Test Set | 82,378 |
| Action | 44,102 |
| Motion | 22,916 |
| Interaction | 15,360 |
| Cross Subject (CS) | 58,882 |
| Cross Action (CA) | 23,496 |
| Occlusion | 7,707 |
| Close-to-Geometry (C2G) | 1,727 |

Table 4.2: Numbers of frames in each test subset. We evaluate performance on different subsets of the test data split by the scripted behavior (Action/Motion/Interaction), subjects that were excluded from the training data (cross-subject) and novel actions (cross-action). Finally, we evaluate on a subset with significant occlusion (Occlusion) and a subset where many joints were near scene geometry (Close-to-Geometry).

## 4.5 Experiments

**Training data:** Our Geometric Pose Affordance (GPA) dataset has 304.8k images of which 82k images are used for held-out test evaluation. In addition, we use the MPII dataset [2], a large scale in-the-wild human pose dataset for training the 2D pose module. It contains

25k training images and 2,957 validation images. For the alternative baseline model (SIM), we use the MPII pre-trained ResNet [190] to detect the 2D key points. We also evaluate performance when using the ground truth 2D human pose, which serves as an upper-bound for the lifting-based method [98].

**Implementation details:** We take a crop around the skeleton from the original $1920 \times 1080$ image and isotropically resize to $256 \times 256$, so that projected skeletons have roughly the same size. Ground-truth target 2D joint location are adjusted accordingly. For ResNet-based method, following [215], the ground truth depth coordinates are normalized to $[0, 1]$. The backbone for all models is ResNet-50 [46]. The 2D heat map/depth map spatial resolution is $64 \times 64$ with one output channel per joint. For test time evaluation, we scale each model prediction to match the average skeleton bone length observed in the training. Models are implemented in PyTorch with Adam as the optimizer. For the lifting-based method we use the same process as above to detect 2D joint locations and train the lifting network using normalized inputs and outputs by subtracting mean and dividing the variance for both 2D input and 3D ground-truth following [98].

**Evaluation metrics:** Following standard protocols defined in [101, 52], we consider two evaluation metrics for experiments: MPJPE (mean per-joint position error) and the 3DPCK (percent correctly localized keypoints) with a distance threshold of 150 mm. In computing the evaluation metrics, root-joint-relative joint locations are evaluated according to the each method original paper evaluation protocol.

**Evaluation subsets:** In addition to the three subsets – Action, Motion, and Interaction – that are inherited from the global split of the dataset based on script contents, we also report test performance on 4 other subsets of the test data: cross-subject (CS), cross-action (CA), occlusion, and close-to-geometry (C2G). These are non-orthogonal splits of the test data

Figure 4.7: We adopt Grabcut [140] and utilize the ground truth (joints, multi-layer depth, and markers) we have to segment subjects from background. If the joints and markers are occluded by the first-layer of multi-layer depth, we set them as background, otherwise they are set as foreground in grabcut algorithm.

which allow for finer characterizations of model performance and generalization in various scenarios: **(1)** CS subset includes clips from held-out subjects to evaluate generalization ability on unseen subjects and scenes; **(2)** CA subset includes clips of held-out actions from same subjects from the training set; **(3)** Occlusion subset includes frames with significant occlusions (at least 10 out of 34 joints are occluded by objects); **(4)** Close-to-geometry subset includes frames where subjects are close to objects (i.e. at least 8 joints have distance less than 175 mm to the nearest surface).

Statistics of these testing subsets are summarized in Table 4.2.

**Ablative study:** To demonstrate the contribution of each component, we evaluate four variants of each model: the baseline models **ResNet / SIM-P / SIM-G** where **G** stands for ground-truth 2D joint input while **P** stands for predicted 2D joint input; **ResNet-E / SIM-P-E / SIM-G-E / PoseNet-E** , models with encoded scene geometry input;

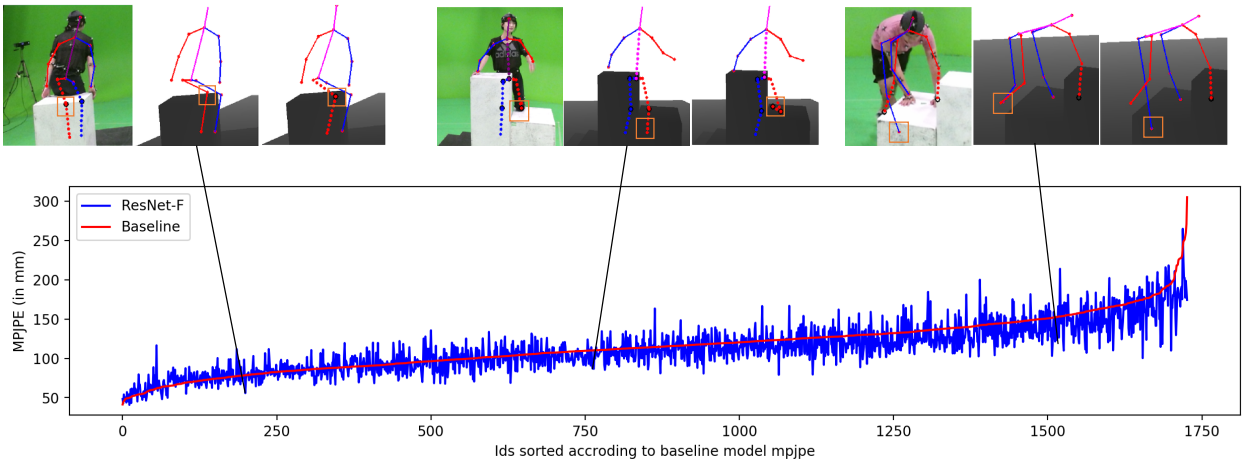



Figure 4.8: Distribution of prediction error (MPJPE) for ResNet-F and the baseline on the close-to-geometry test set. Examples are sorted in increasing order of baseline MPJPE (red) with corresponding ResNet-F performances (GCL + encoding, in blue). We also highlight 3 qualitative results, from left to right: (a) case shows ResNet-F improve over the baseline with respect to the depth prediction. (b,c) cases show ResNet-F improves over the baseline in all $x, y, z$ axes. Furthermore, (b) demonstrates ResNet-F can even resolve ambiguity under heavy occlusions with the aid of geometry information. We show the image with the estimated 2D pose (after cropping), 1st layer of multi-layer depth map and whether the joint is occluded or not. **Legend:** hollow circles: occluded joints; solid dots: non-occluded joints; dotted lines: partially/completely occluded body parts; solid lines: non-occluded body parts.

**ResNet-C / SIM-P-C / SIM-G-C / PoseNet-C**, the models with geometric consistency loss (GCL); **ResNet-F / SIM-P-F / SIM-G-F /PoseNet-F**, our full model with both encoded geometry priors and GCL.

### 4.5.1 Baselines

To evaluate the difficulty of the GPA and provide context, we trained and evaluated a variety of recently proposed architectures for pose estimation including: DOPE [186], Simple baseline [98], ResNet-Baseline [215], PoseNet [107], and I2L [108]. As data and code for training DOPE was not available, we evaluated their released model. To account for systematic differences in the body joint definitions, we utilized the average of hip joints as the DOPE coordinate origin (H36M-based models typically use the pelvis root joint as the origin). For

| MPJPE | Baseline | ResNet-E | ResNet-C | ResNet-F |
|---|---|---|---|---|
| Full | 96.6 | 94.6 | 95.4 | 94.1 |
| Action | 97.2 | 95.8 | 96.6 | 95.1 |
| Motion | 99.6 | 97.0 | 97.9 | 96.5 |
| Interaction | 89.7 | 87.5 | 88.3 | 87.4 |
| CS | 99.4 | 98.1 | 98.8 | 97.8 |
| CA | 89.2 | 85.8 | 86.7 | 85.6 |
| Occlusion | 120.5 | 116.1 | 117.9 | 115.1 |
| C2G | 118.1 | 113.2 | 116.3 | 111.5 |

Table 4.3: Prediction error (MPJPE) for ResNet-based models over the full test set as well as different test subsets. Our proposed geometric encoding (ResNet-E) and geometric consistency loss (ResNet-C) each contribute to the performance of the full model (ResNet-F). Most significant reductions in error are for subsets involving significant interactions with scene geometry (Occlusion,C2G)

| MPJPE | Baseline | PoseNet-E | PoseNet-C | PoseNet-F |
|---|---|---|---|---|
| Full | 62.8 | 62.3 | 62.5 | 62.0 |
| C2G | 69.8 | 69.1 | 69.0 | 68.5 |
| Full | 78.8 | 78.5 | 78.2 | 78.1 |
| C2G | 91.9 | 91.4 | 91.6 | 89.4 |

Table 4.4: Prediction error (MPJPE) for ResNet-based models over the full test set as well as different test subsets. Our proposed geometric encoding (PoseNet-E) and geometric consistency loss (PoseNet-C) each contribute to (PoseNet-F).

| Method | Full set | C2G |
|---|---|---|
| Lifting [98] | 91.2 | 112.8 |
| ResNet-Baseline [215] | 96.6 | 118.1 |
| PoseNet [107] | <u>62.8</u> | <u>70.7</u> |
| I2L [108] | 68.1 | 80.4 |
| DOPE [186] | 126.0 | 150.2 |
| PoseNet (masked background) | 64.4 | 78.7 |
| Ours (PoseNet-F) | **62.0** | **68.9** |

Table 4.5: We evaluated MPJPE (mm) for several recently proposed state-of-the-art architectures on our dataset. All models except DOPE were tuned on GPA training data. We also trained and evaluated PoseNet on masked data (see Fig. 7) to limit implicit learning of scene constraints.

| Dataset tested on / trained on | GPA | H36M |
|---|---|---|
| H36M [52] | 118.8 | 61.4 |
| GPA | 62.8 | 110.9 |
| SURREAL [171] | 126.2 | 142.4 |
| 3DPW [173] | 125.5 | 132.5 |
| 3DHP [101] | 150.9 | 154.0 |

Table 4.6: PoseNet models trained on our GPA dataset generalize well to other test datasets, outperforming models trained on H36M despite $\sim 30\%$ fewer training examples [182]. We attribute this to the greater diversity of poses, occlusions and scene interactions present in GPA.

| PCK3D | Baseline | ResNet-E | ResNet-C | ResNet-F |
|---|---|---|---|---|
| Full | 81.9 | 82.5 | 82.3 | 82.9 |
| Action | 81.4 | 81.8 | 81.6 | 82.0 |
| Motion | 80.7 | 81.5 | 81.6 | 82.0 |
| Interaction | 85.2 | 86.0 | 85.7 | 86.1 |
| CS | 81.3 | 81.7 | 81.5 | 82.0 |
| CA | 83.6 | 84.7 | 84.5 | 84.8 |
| Occlusion | 72.2 | 73.9 | 73.7 | 74.2 |
| C2G | 71.4 | 73.7 | 72.1 | 74.7 |

Table 4.7: Localization accuracy (PCK3D) follows similar trends to the mean errors reported in Table 4.3.

the other architectures, we train and test on the GPA dataset following the original authors' hyperparameter settings. The results are illustrated in Table 4.5. We can see a range of performance across different architectures, ranging from 62.8 to 91.2 mm in MPJPE metric. Our full model built on the PoseNet architecture achieves the lowest estimation error.

We break down the performance of the ResNet-based joint regression baseline on different subsets of data in Table 4.3. We also list the corresponding PCK3D in Table 4.7, which follows a similar pattern. The motion, occlusion and close-to-geometry subsets prove to be the most challenging as they involve large numbers of frames where subjects interact with the scene geometry.

**Cross-dataset Generalization** We find that pose estimators show a clear degree of overfitting to the specific datasets on which they are trained on [182]. To directly verify whether the model trained on GPA generalizes to other datasets, we trained the high-performing PoseNet architecture using GPA and MPII [2] data, and tested on several popular benchmarks: SURREAL [171], 3DHP [101], and 3DPW [173]. To evaluate consistently across test datasets, we only consider error on a subset of 14 joints which are common to all. The MPJPE (mm) is illustrated in Table 4.6. We can see the model trained on GPA generalizes to other datasets with similar or better generalization performance compared to the H36M trained variant. This is surprising since H36M train is roughly 30% larger. We attribute this to the greater diversity of scene interactions, poses and occlusion patterns available in GPA train.

### 4.5.2 Effectiveness of geometric affordance

From Table 4.3 we observe that incorporating geometric as an input (ResNet-E) and penalizing predictions that violate constraints during training (ResNet-C) both yield improved performance across all test subsets. Not surprisingly, the full model (ResNet-F) which is

trained to respect geometric context provided as an input achieves the best performance. We can see from Table 4.3 that the full model, ResNet-F decreases the MPJPE by 2.1$mm$ over the whole test set. Among 4 subsets, the most significant improvement comes on the occlusion and close-to-geometry subsets. Our geometry-aware method decreases MPJPE in occlusion and C2G set by 5.4mm / 6.6mm and increase the PCK3D about 2% / 3%. Similar results hold for the SIM model. The MPJPE is reduced when using either the predicted (SIM-P-F) or ground-truth 2D joint locations (SIM-P-F) by 3mm and 3.6mm respectively (PCK3D improves 1.2% and 1.1%). The improvement from SIM-G model is overall larger than SIM-P model due to the more accurate 2D location and better geometry information provided to the network.

**Controlling for Visual Context** One confounding factor in interpreting the power of geometric affordance for the ResNet-based model is that while the baseline model doesn't use explicit geometric input, there is a high degree of visual consistency between the RGB image and the underlying scene geometry (e.g., floor is green, boxes are brighter white on top than on vertical surfaces). As a result, the baseline model may well be implicitly learning some of the scene geometric constraints from images alone and consequently decreasing the apparent size of the performance gap.

To further understand whether the background pixels are useful or not for 3D pose estimation, we utilize Grabcut [140] to mask out background pixels. Specifically, we label the pixel belonging to markers, joints that are not occluded by the first-layer of multi-layer depth map as foreground, and occluded ones as background. Additionally, we dilate the skeleton constructed by all the joints and markers, use the inverse area as background area. We send these labels together with the image to OpenCV implementation Grabcut and get the foreground mask. We set the background color as green for better visualization as shown in Fig 4.7. We use the model [107], and train and test on the masked background images.

| MPJPE (mm) | Predicted Root | | Ground Truth Root | |
|---|---|---|---|---|
| | C2G | Full | C2G | Full |
| ResNet | 118.1 | 96.5 | | |
| ResNet-E | 116.0 | 95.4 | 113.2 | 94.6 |
| ResNet-F | **115.1** | **94.7** | **111.5** | **94.1** |
| SIM-P-B | 112.8 | 91.2 | | |
| SIM-P-E | 106.9 | 89.2 | 105.2 | 89.1 |
| SIM-P-F | **105.1** | **88.9** | **104.2** | **88.2** |
| SIM-G-B | 79.8 | 68.2 | | |
| SIM-G-E | 76.3 | 65.3 | 74.2 | 64.8 |
| SIM-G-F | **74.9** | **65.0** | **72.8** | **64.6** |

Table 4.8: The root joint depth is needed to offset the multi-layer depth map when encoding the scene geometry for relative pose estimation. Inaccurate root joint prediction limits but does not eliminate the benefits of the geometric encoding.

We observe increased error on C2G from 70.7 mm to 78.7 mm MPJPE, which suggests that baseline models do take significant advantage of visual context in estimating pose.

**Errors by joint type:** We partition the 16 human joints into the limb joints which are more likely to be interacting with scene geometry (out group) and the torso and hips (in group). The performance on these two subsets of joints as well as individual joints is illustrated for the SIM model in Table 4.9. This verifies our assumption that limb joint estimation (wrist, elbow, knees, ankles) benefits more from incorporating geometric scene affordance.

**Error in predicted root joint:** Since our models predict joint depths relative to the root joint, it is necessary to offset the multi-layer depth map values when encoding them as input. To make our evaluation more realistic, we also evaluated models using predicted root joint locations instead of using the ground-truth. To estimate the (absolute) root joint depth, we utilize the model and training procedure from [107] which estimates root joint depth based on the person bounding-box size and image features. This yields a mean root position error (MRPE) of 136.6mm with a mean z-coordinate (depth) error of 116.8mm and x- and y-coordinate errors of 41.6mm and 35.2mm respectively. Table 4.8 shows the result of

| MPJPE (mm) | SIM-G | SIM-G-F | SIM-P | SIM-P-F |
|---|---|---|---|---|
| righthip | 17.1 | 15.5 | 22.5 | 20.7 |
| lefthip | 17.3 | 15.8 | 22.9 | 21.4 |
| spine1 | 48.3 | 44.9 | 63.5 | 62.4 |
| head | 55.1 | 51.7 | 70.4 | 69.0 |
| rightshoulder | 58.5 | 54.1 | 75.9 | 74.2 |
| leftshoulder | 61.0 | 56.7 | 78.7 | 75.4 |
| leftknee | 64.1 | 60.6 | 88.9 | 84.9 |
| rightknee | 64.6 | 61.3 | 91.8 | 87.2 |
| rightelbow | 81.4 | 75.1 | 108.5 | 103.6 |
| leftforeelbow | 84.5 | 81.8 | 104.1 | 102.9 |
| neck | 86.1 | 81.3 | 102.0 | 98.8 |
| rightankle | 86.6 | 83.2 | 127.5 | 122.1 |
| leftankle | 88.9 | 86.2 | 131.0 | 125.8 |
| rightwrist | 102.5 | 96.1 | 140.1 | 135.7 |
| leftwrist | 107.1 | 104.8 | 138.5 | 138.3 |
| in-group | 49.1 | 45.7 | 62.4 | 60.3 |
| out-group | 85.0 | 81.1 | 116.3 | 112.6 |
| all joints | 68.2 | 64.6 | 91.2 | 88.2 |

Table 4.9: Performance of the lifting network-based model [98] broken down by individual joints and joint subsets. Baseline prediction error is higher for extremities (e.g., wrists and ankles) which are inherently more difficult to localize. These same joints typically show the largest reduction in error from introducing geometric context.

using this predicted root joint depth during encoding to offset the multi-depth map. Using predicted depth results in a loss of performance of about 1% over the three methods (with the largest effect for ResNet) but does not eliminate the benefits of geometric context.

**Computational Cost:** We report the average runtime over 10 randomly sampled images on a single 1080Ti in Table 4.10. Timings for SIM do not include 2D keypoint detection. For comparison, we also include the run time for the PROX model of [44] which uses an optimization-based approach to perform geometry-aware pose estimation.

**Qualitative results:** We show qualitative examples that high-light interaction with geometry in Fig 4.8 along with the distributions of the mean prediction error for the baseline and ResNet-F model over the close2geometry subset. The geometry aware model is able

| Method | Average Run Time |
|---|---|
| SIM [98] | 0.57 ms |
| SIM-F | 0.64 ms |
| ResNet [215] | 0.29 s |
| ResNet-F | 0.36 s |
| PROX [44] | 47.64 s |

Table 4.10: We compare the running time for our baseline backbone, our method, and another geometry-aware 3D pose estimation method PROX [44] averaged over 10 samples evaluated on a single GPU.

to show most improvement for hard examples where the baseline error is large. Further visualization of model predictions along with scene geometry encodings are shown in Fig 4.9. These examples demonstrate that ResNet-F has better accuracy in both $xy$ localization and depth prediction and is often able to resolve ambiguity under heavy occlusion where the baseline fails.

## 4.6 Discussion and Conclusion

In this work, we introduce a large-scale dataset for exploring geometric pose affordance constraints. The dataset provides multi-view imagery with gold-standard 3D human pose and scene geometry, and features a rich variety of human-scene interactions. We propose using multi-layer depth as a concise camera-relative representation for encoding scene geometry, and explore two effective ways to incorporate geometric constraints into training in an end-to-end fashion. There are, of course, many alternatives for representing geometric scene constraints which we have not yet explored. We hope the availability of this dataset will inspire future work on geometry-aware feature design and affordance learning for 3D human pose estimation.

Broadly speaking, our techniques for encoding geometry yielded only modest reductions joint localization error ($\sim 2-6\%$ depending on the base model). We might have hoped for greater gains, but we expect that even the baseline models are implicitly learning something about

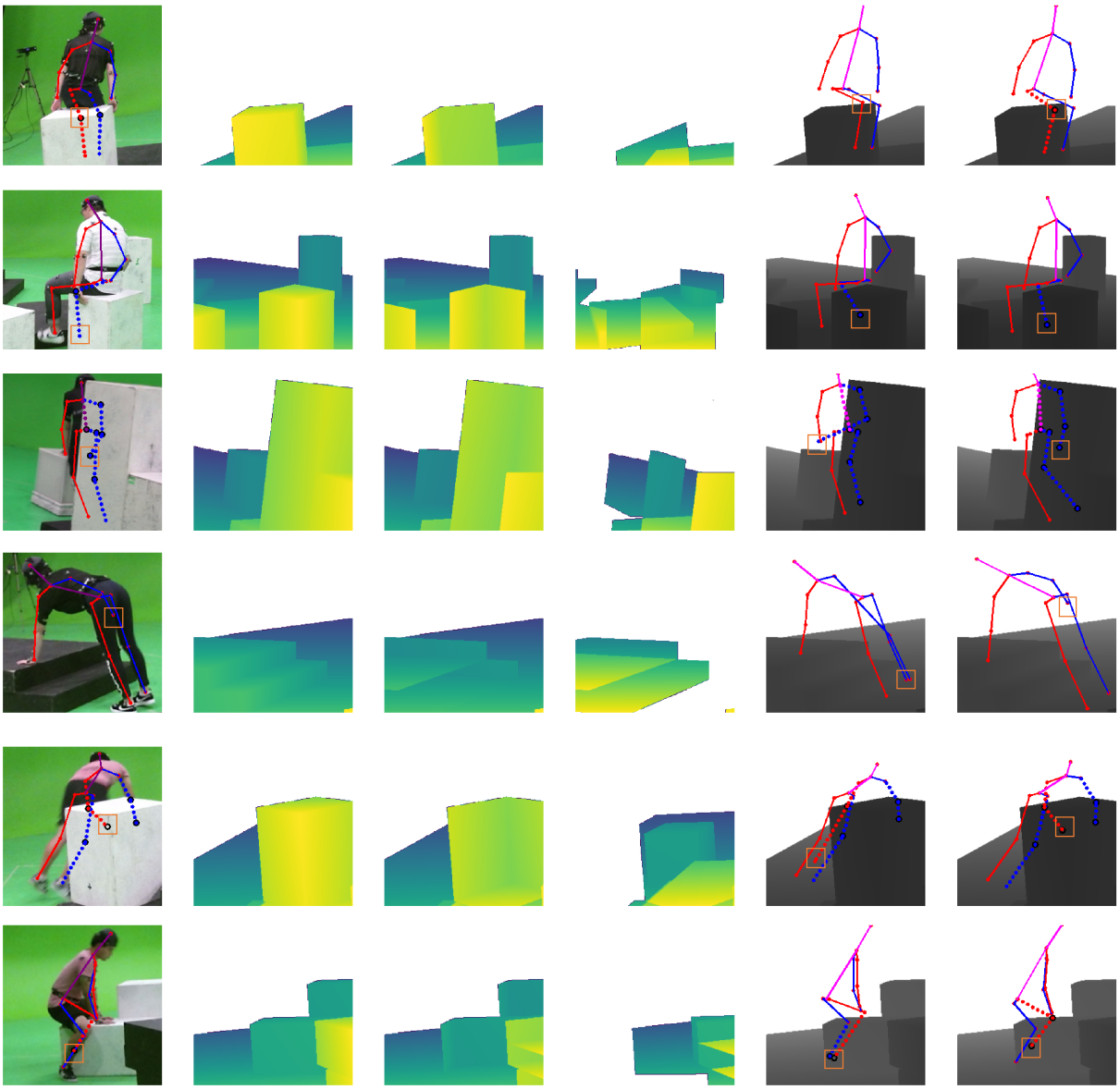


Figure 4.9: Visualization of the input images with the ground truth pose overlaid in the same view (blue and red indicate right and left sides respectively). Columns 2-4 depict the first 3 layers of multi-layer depth map. Column 5 is the baseline model prediction overlaid on the 1st layer multi-layer depth map. Column 6 is the ResNet-F model prediction. The red rectangles highlight locations where the baseline model generates pose predictions that violate scene geometry or are otherwise improved by incorporating geometric input.

scene constraints that are common across our dataset. Indeed, masking out the background yielded an $\sim 11\%$ increase in baseline error. There has been substantial success in training models that predict scene depth (2.5D) from monocular RGB inputs [25, 15] as well as full

3D representations such as voxels [147, 166] or multilayer depth [144]. This suggests that when geometric supervision is available, it may be useful to explore training systems that jointly estimate scene structure and 3D human pose in a multi-task setup.

In our experiments we focused on a setting where the scene geometric constraints were available as input and highly accurate. While such prior knowledge is not available in general (e.g., for a random photo on the web), we believe such data is readily accessible in many practical scenarios. The successful development of robust structure from motion, SLAM, and specialized stereo or time-of-flight depth sensors makes geometric scene information increasingly prevalent and easy to acquire. Assuming known camera and scene geometry as input appears practical in commercial applications where, e.g. robots navigate a well-mapped environment interacting with people or fixed cameras monitor human activity in a static workspace. We expect finding better techniques to incorporate such "side information" will offer a way to improve cross-scene/cross-dataset generalization and avoid some of the common over-fitting we currently observe when training and testing on individual datasets.

# Chapter 5

# Combining Model-based and Nonparametric Approaches for 3D Human Body Estimation

## 5.1 Introduction

The 3D estimation of the human body pose and shape from a monocular image is a fundamental task for various applications such as VR/AR, virtual try-on, metaverse and animations. It is challenging mostly due to the depth ambiguity and lack of evidence from single image. There are several ways to solve this ambiguity such as leveraging multi-view or video data to fuse image evidence from more images and infer occluded parts. For the case of single images, researchers used parametric models such as SMPL [92] to fit 2D image evidence [68] or use human pose prior [61, 116, 64] to penalize problematic human pose / mesh prediction in combination with modern deep learning techniques. However, these model-based methods are prone to produce corrupted results when severe occlusion happens.

Nonparametric methods use non-compressed representations like voxels [118], heatmaps [108] and joint location [151, 88, 89] as the target for modern deep learning. However, to estimate dense meshes they are computationaly expensive and consume lots of memory. They either use integral methods to estimate normalized joint location [108] or simplify meshes [89] to reduce the number of vertices. Without post-processing, these methods also generate qualitatively non-pleasing results. The dense correspondence methods [206, 205, 207], which are based on template SMPL human mesh surface and have been proven for various tasks.

Connecting nonparametric methods and model-based methods is hard due to the difficulty in localizing the corresponding feature. [38, 209, 108] utilize bounding boxes or keypoints location to find the related features to estimate necessary SMPL parameters. While [65, 88] learn the feature-parameter correspondence (attention) implicitly through neural networks. [205, 208] consider the correspondence between the mesh representation and pixel representation based on human surface mapping (UV coordinate system). However, they estimate the SMPL parameter through a light weight FC network and treat this simple optimization process as a post process. Their methods also do not convey the advantages of nonparametric methods such as robustness to occlusion.

To leverage the advantages from both worlds, we propose a 3D human body estimation framework that consists of three modules: Dense Map Prediction module ($DMP$), Inverse Kinematics module ($IK$) and UV Inpainting module ($UVI$). $DMP$ explicitly predicts per-pixel human 3D joint location, 3D surface location in root relative coordinates, 3D displacement between the joint location and surface location, and also predicts UV coordinates which represent the human surface in a 2D grid. This module is robust to partial occlusion when predicting joint, as all the image evidence belongs to this part will contribute to the prediction explicitly. $IK$ module connects the nonparametric prediction to model-based method. We first warp the DMP dense prediction to UV space and get the joint prediction based on the part-segmentation in UV space. Then we use a two-stage multi-layer perceptron, where the

first stage inpaints and refines the joint prediction, while the second stage estimates SMPL parameters and eventually produces a posed mesh. With all the predictions in UV space from *DMP* and *IK*, *UVI* inpaints and refines the 3D body pose and mesh in UV space.

In summary, our contributions are three fold:

• We propose a 3D body estimation framework from single image that seamlessly leverages the best of the both worlds (model-based and nonparametric).

• The method is robust to occlusions and can self-correct wrong poses from Dense Map Prediction module.

• We achieve state-of-the-art performance on H36M and 3DOH datasets.

## 5.2 Related Work

**3D human shape estimation from monocular images** SMPL [92] has been widely used for 3D human mesh reconstruction. To boost its power in practice, a number of deep learning frameworks have been proposed by using SMPL as regression targets [61, 68, 116, 205, 108]. [61] regresses SMPL parameters directly from input images by end-to-end training. Following this research direction, [108] add spherical Gaussian attention joint based on initial joint estimation, and the use the the attended feature to learn the vertices location. [68] combine learning and optimization[116] in the same framework but cannot handle occlusions. [205] uses the template UV mapping from SMPL and transforms 3D mesh reconstruction to decomposed UV estimation and position map inpainting problems. However, the way to get 3D human joint from SMPL mesh is based on the pre-trained joint regressor, which will induce intrinsic errors and usually does not generalize to other datasets.

**3D human pose estimation from monocular images** Deep learning approaches have shown success in regressing 3D pose from a single image [107, 139, 190, 98, 213, 118]. Basically,

most current models can be categorized into two frameworks. The first is to directly estimate 3D pose from images, based on volumetric representation [118, 107]. But these approaches may involve in high memory consumption and complex post-processing steps. Based on the explosive improvement in 2D pose estimation [190], another framework is to estimate 2D pose from images and then lift 2D pose to 3D pose [213, 98]. Since these approaches take 2D joint locations as input, 3D human pose estimation simply focuses on learning depth of each joint. This releases learning difficulty and leads to better 3D pose. However, there are few methods on systematically handling occlusion in the first framework while the second framework cannot recover information if the joint detector fails. Additionally, how to get human surfaces from the joint prediction remains a problem.

**Inverse Kinematics** The inverse kinematics (IK) problem has been extensively studied in robotics [4, 187] and graphics [34] and its techniques have been used in 3D human pose estimation [172, 81, 65, 222, 223]. Numerical solutions [4, 34, 187] rely on time-consuming iterative optimization. [172] uses temporal sequence to resolve IK ambiguity. [81] decomposes the IK rotation to the product of swing rotation and twist rotation and solve swing rotation analytically from predicted joint locations. Feed forward solution like [223, 222] propose BodyIKNet to regress SMPL [92] pose and shape parameters from 3D joint location, However, it leads to a sub-optimal solution when partial occlusion happens.

**Occlusion** [153] presented a systematic study of various types of synthetic occlusions in 3D human pose estimation from a single RGB image. Since synthetic data can not fully depict the real occlusion, [35] learns from real data and uses grammar models with explicit occluding templates to reason about occluded people. To avoid specific design for occlusion patterns, [32] presents a method for modeling occlusion that aims at explicitly learning the appearance and statistics of occlusion patterns. They also synthesizes a large corpus of training data by compositing segmented objects at random locations over a base training

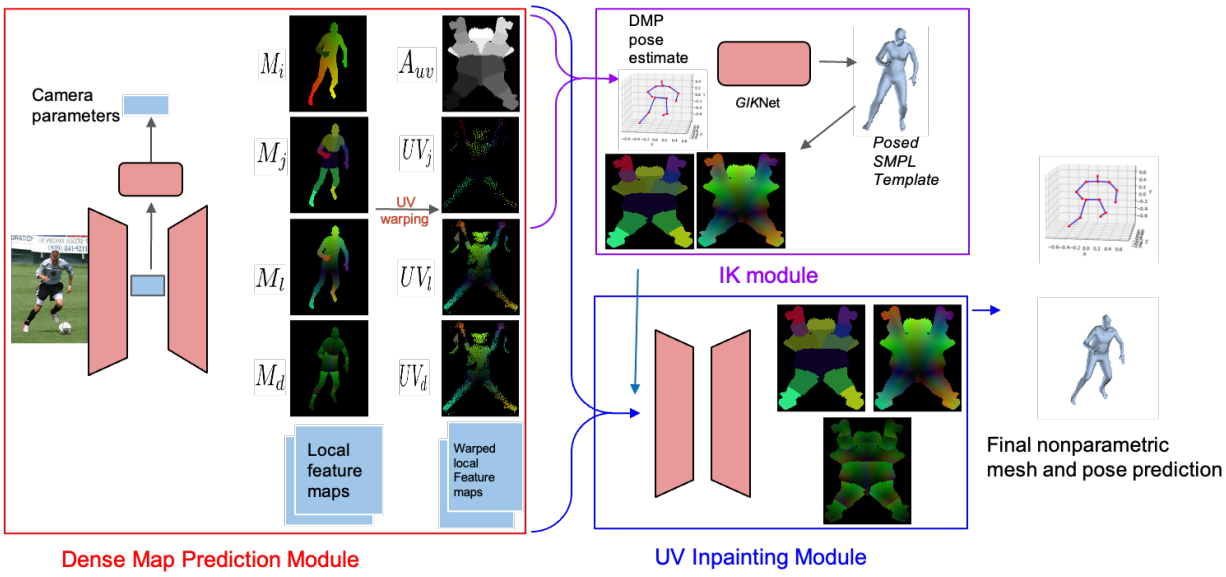



Figure 5.1: Our 3D body estimation framework consists of three part: Dense Map Prediction module ($DMP$), Inverse Kinematics and SMPL module ($IK$) and UV Inpainting Module ($UVI$).

image. [19] utilizes a cylinder model and confidence maps to filter out the occluded joints and uses flow warped joint in the same video to approximate the missing joints. [128] integrates depth information about occluded objects into 3D pose estimation. To provide full-geometry information to handle occlusion scenarios, [180] and [44] provide 3D scene geometry as multi-layer depth maps or signed distance fields into the inference stage. [136] proposes a simple but effective self-training framework to adapt the model to highly occluded observations. To fully utilize the holistic human body model (e.g. SMPL [92]), [210] represents the target SMPL human mesh as UV location map and converts the full-body human estimation as an image inpainting problem. However, these frameworks either rely on nonparametric estimation or pure model-based regression, how to leverage the best of both worlds seamlessly remain an unexplored problem.

## 5.3 Method

As shown in Figure 5.1, our framework consists of three consecutive modules, including a dense map prediction module ($DMP$), which extract dense semantic maps (e.g. 3D joint location, surface location and their displacements) and correspondence UV position, an inverse kinematics and SMPL module ($IK$), which inpaint 3D joint location and estimate the smpl parameters, as well as a UV map inpainting module, which estimate the final joint location and mesh location in UV space.

### 5.3.1 Dense Map Prediction Module

Our dense map prediction module is an encoder-decoder architecture and is used to extract the IUV images $M_i$, as well as dense semantic maps including dense joint map $M_j$, dense location map $M_l$ and dense displacement maps $M_d$. They are further illustrated in Fig. 5.2. $M_i$ is generated from the continuous UV map from [205], it is continuous in both image space and UV space, thus, easier to learn compared with original UV map [92]. It is used to convert the dense local features as well as these semantic maps to UV space. For location map $M_l$, it represents the position of each vertices from the SMPL human mesh surface in root-relative coordinates. To construct $M_l$ groundtruth, we first use the SMPL model, SMPL parameters and camera parameters to generate the vertices location in root-relative coordinate, and generate the full UV space location map $UV_l$ using barycentric interpolation (The mesh faces correspondence is defined by [205]). After that we use the $M_i$ to fetch values from $UV_l$ to get the dense location map in image space. For the generation of dense joint map $M_j$, we first rely on T-pose SMPL mesh and assign each vertex to the nearest joints (14 LSP joints setting), after which we use barycentric interpolation to get the UV space assignment, and further refine the assignment by make it symmetric in UV space (e.g. left hip and right hip has symmetric shape in UV space, as illustrated in Fig 5.3). We term the part assignment

in UV space as $A_{uv}$ . After setting the assignment in UV space, we use the $M_i$ to query values from $UV_j$ to get the dense joint map in image space. $UV_j$ stores the root-relative joint location. We define displacement as the residual between vertex location and the assigned joint location, thus $UV_d = UV_l - UV_j$ and $M_d = M_l - M_j$. As our human bodies are usually left-right symmetric (e.g. left hand has symmetric shape with right hand and the size and the distance between joint and surface is almost the same.), the magnitude of left part and right part of $UV_d$ should be the same.

These semantic maps are aligned with the human in the images. Thus we are able to train a encoder-decoder network to estimate directly from image space. Dense image space joint prediction shares the similar flavor with [193, 115].

The objective for the dense map prediction module is

$$\ell_{DMP} = \ell_{M_i} + \ell_{M_l} + \ell_{M_j} + \ell_{M_d} \tag{5.1}$$

$\ell_{M_i}$ is composed of two parts: a binary mask loss $\ell_{M_ib}$ of human body, which distinguishes pixels from those at the background, and the human pixels. The loss function of $\ell_{M_ib}$ is binary cross entropy loss. our CNN further outputs the UV coordinates and uses L1 loss $\ell_{M_{iuv}}$ .

$$\ell_{M_i} = \ell_{M_{ib}} + \ell_{M_{iuv}} \tag{5.2}$$

For $\ell_{M_l}$, $\ell_{M_j}$ and $\ell_{M_d}$, we use L1 loss to directly regress the real value. As these values are already in root-relative coordinate and in unit meters, thus their data range is $-1$ to $+1$, we

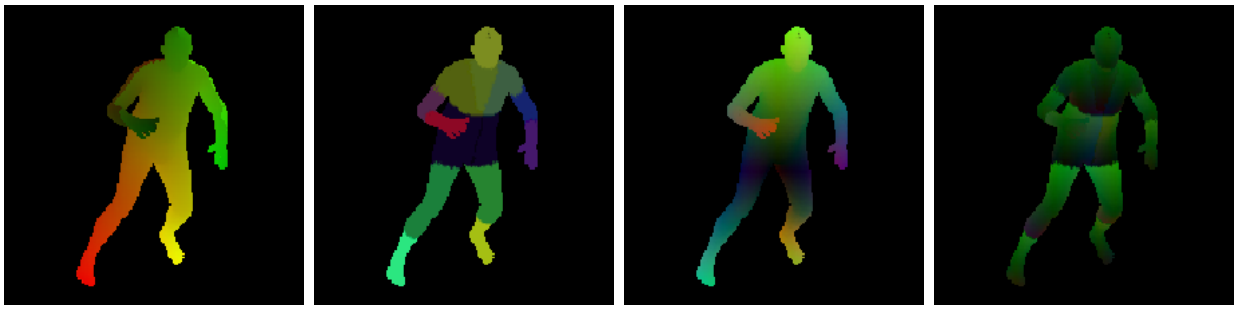


Figure 5.2: Semantic maps aligned with image space. From left to right: IUV image $M_i$, Dense jointmap $M_j$, dense location map $M_l$ and dense displacement map $M_d$. (Best viewed in Color)

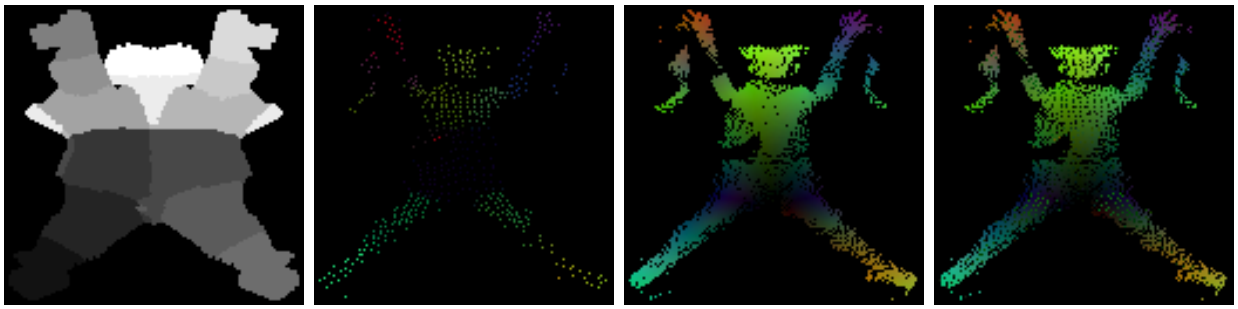


Figure 5.3: Warped Images in UV space based on IUV images $M_i$. From left to right: Part segmentation in UV space $A_{uv}$, UV space jointmap $UV_j$, UV space location map $UV_l$ and UV space displacement map $UV_d$. (Best viewed in Color)

do not further normalize them.

Our dense map prediction module not only predicts these semantic maps, but also extracts both global feature to estimate camera parameter and local feature for the UV impainting module.

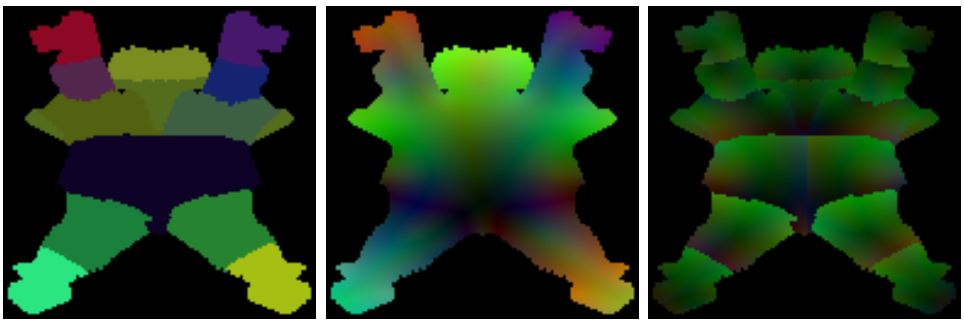


Figure 5.4: Full groundtruth in UV space. From left to right: UV space jointmap $UV_j$, UV space location map $UV_l$ and UV space displacement map $UV_d$. (Best viewed in Color)

### 5.3.2 Inverse Kinematics Module

**Estimate Joint Location from *DMP*** After warping the semantic maps $(M_l, M_j, M_d)$ from image space to uv space, we get the incomplete uv joint map $UV_j$. Based on the uv space joint assignment $A_{uv}$ (as shown in Fig 5.3), we aggregate the dense prediction $UV_j$ for each joint and average them if they are not fully occluded. Thus we have a coarse prediction for each joint $J_{initial}$.

**Joint Inpaint and Refine Module** Even though each human pixel contributes to joint prediction, there are still cases where some joints have no assigned vertex/pixel available from the image evidence. Thus we propose the joint inpainting module to inpaint these missing joints. This network is pretty flexible and can be MLP [98], GCN [213] or even modern transformers [88]. For the ease of implementation we use simple multi-layer perceptron. Our joint inpainting net is inspired by [98], which is simple, deep and a fully-connected network with six linear layer with 256 output features. It includes dropout after every fully connected layer, batch-normalization and residual connections. The model contains approximately 400k training parameters. The goal of this network is not only to inpaint the joints but also to refine the joints prediction that is not occluded. It takes the $J_{initial}$ as input and the output of the network is the joint in root-relative coordinates $J_{refine}$. We use L1 loss $L_{ji}$ to train joint inpaint and refine module. The structure of the joint inpainting and refine module is shown in Fig 5.5.

**Inverse Kinematics Module** After getting the sparse 3D human keypoints. We want to repose the template SMPL meshes based on the predicted joints location. To solve this problem we leverage inverse kinematics (IK). Typically, the IK task is tackled with iterative optimization methods [4, 34, 187], which requires a good initialization, more time and case-by-case optimization method. Here we propose a global inverse kinematics neural

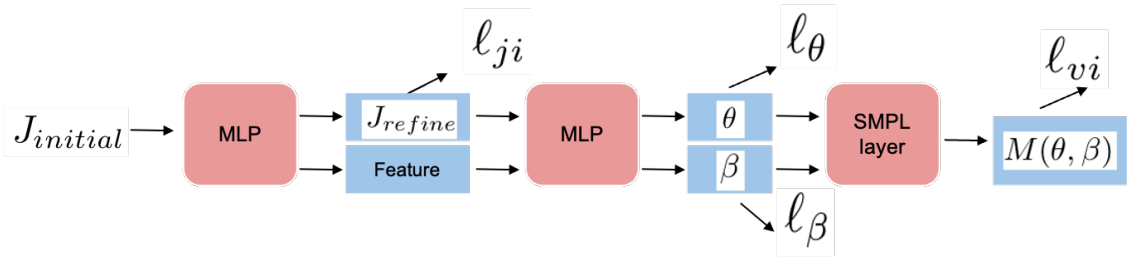



Figure 5.5: Structure of *GIKNet*. (Best viewed in Color)

network *GIK-Net*. This network is constructed by the basic fully connected neural network module with residual connection, batch normalization and relu activation similar to [98]. In particular, *GIK-Net* takes the refined keypoint coordinates $J_{refine}$ in root-relative space and outputs joint rotations $\theta$ and $\beta$ which serve as the input for SMPL layer. As we also use the Mocap dataset (AMASS [95], SPIN[68] and AIST++ [82]), our *GIK-Net* can learn the realistic distribution of human kinematics rotation and human shape implicitly. The use of the additional Mocap dataset serves the same purpose as the factorized adversarial prior [61], variational human pose prior [116] and motion discriminator [64]. We use L1 loss $L_\theta$ and $L_\beta$ to train *GIK-Net*. The structure of *GIK-Net* is shown in Fig 5.5.

**SMPL revisits and Reposing Module** SMPL [92] represents the body pose and shape by pose $\theta \in R^{72}$ and shape $\beta \in R^{10}$ parameter. Here we use the gender-neural shape model following previous work [65, 61, 68]. Given these parameters, the SMPL module is a differentiable function that outputs a posed 3D mesh $M(\theta, \beta) \in R^{6890\times3}$. The 3D joint locations $J_{3D} = WM \in R^{J\times3}$, while J are computed with a pretrained linear regressor $W$. After getting the $\theta$ and $\beta$ from the *GIK-Net* we send them to SMPL layer to get the body mesh prediction.

We also augment the joints input for *GIK-Net* from Mocap dataset with guassian noise and random synthetic occlusion (30%). The augmentation helps our *GIK-Net* generalize to more realistic noisy input. We use L1 loss $L_{vi}$ to train the mesh prediction from SMPL module.

The objective for the inverse kinematics and smpl module is

$$\ell_{IK} = \ell_\theta + \ell_\beta + \ell_{ji} + \ell_{vi} \tag{5.3}$$

### 5.3.3 UV Inpainting Module

The goal of UV inpainting module is to regress 3D joint and mesh location directly based on the feature / semantic output $(UV_l, UV_j, UV_d)$ from *DMP* and semantic output $(UV_l, UV_j, UV_d)$ from *IK*.

**Inevitable Fitting Error introduced by SMPL model and Joint regressor** The advantage of directly regressing joint/mesh location over model-based method is that model-based method will introduce intrinsic fitting error. Specifically, if we use the SMPL layer, groundtruth SMPL parameters (from Mosh), and the joint-regressor [68] to obtain fitted joint for the whole Human3.6M dataset, we get average fitting error as 24.1 mm (MPJPE) when compared with the Human3.6M joint from Mocap system. This means that even if we predict perfect SMPL mesh we still have about 24.1 mm fitting error. Thus we argue directly training and estimating joint location from UV space is a better alternative solution.

**UV inpainting module** After getting the refined joint location $J_{refine}$ from *IK* module, we distribute the refined joint location in UV space based on UV space joint assignment map $A_{uv}$ and generate refine UV joint map $UV_{jrefine}$. We also have the reposed template mesh and the corresponding reposed UV location map $UV_l$ (through barycentric interpolation). Additionally, we have features $UV_f$, location map $UV_l$, joint map $UV_j$ and displacement $UV_d$ from *DMP*. We combine the best of both worlds ( *DMP* and *IK*) feature through aggregation

and send it to our UV inpainting module. The UV inpainting module is a light UNet with skip connections.

For the training of the UV inpainting module, we have

$$\ell_{map} = \|\hat{UV}_{map} - UV_{map}\|_1 \tag{5.4}$$

Note the 'map' represents location map, joint map and displacement map in uv space. Addtionally, we have 3D joints and 2d joint loss based on the predicted camera parameter. Our camera parameters consist of scale and offset parameter to map the xy in $J_{3D}$ to $J_{2d}$.

$$\ell_{j3D} = \|\hat{J}_{3D} - J_{3D}\|_1 \tag{5.5}$$

$$\ell_{j2d} = \|\hat{J}_{2d} - J_{2d}\|_1 \tag{5.6}$$

As we know, the distance between the human surface to the joints are left-right symmetric, thus we also apply symmetric loss on the magnitude of displacement.

$$\ell_{dismag} = \|\|\hat{UV}_d\| - \|\hat{UV}_d^{flip}\|\|_1 \tag{5.7}$$

To align the predicted mesh surface with image aligned IUV images $M_i$, we also adopt consistent loss from [205]. It is enabled by the camera parameter predicted by our model (scaling and offset parameter).

The objective for the uv inpainting module is

$$\ell_{UVI} = \ell_{dismag} + \ell_{j2d} + \ell_{j3D} + \ell_{map} + \ell_{con} \tag{5.8}$$

Thus we have all the losses as

$$\ell_{all} = \ell_{DMP} + \ell_{IK} + \ell_{UVI} \tag{5.9}$$

**Inference** We do inference of 3D joint location from $UV_j$ and based on the uv assignment $A_{uv}$ for each joint. We average all the prediction for the specific joints if this pixel prediction is valid. For human mesh prediction we use the barycentric interplatation from the UV space location map $UV_l$.

### 5.3.4 Implementation Details

The proposed framework is trained on the ResNet-50 [46] backbone pre-trained on ImageNet. It takes a 224 × 224 image as input, and input resolution for *UVI* is 64 × 64 and the output resolution is 128 × 128. We train three modules separately. We first train our *DMP*, and based on the output of *DMP* and Mocap data we train our *IK*; We finally fix and concat *DMP* and *IK*, and train *UVI* module. We apply synthetic occlusion [154] when train *DMP*. The training data is augmented with randomly scaling, rotation, flipping and RGB channel noise. We use the Adam optimizer. The training data for each module is illustrated in table 5.1.

| Stages | Training Datasets |
|---|---|
| *DMP* | H36M, MPI-INF-3DHP, MPII, COCO, LSP |
| *IK* | H36M, MPI-INF-3DHP, AMASS, AIST++ |
| *UVI* | H36M, 3DOH |

Table 5.1: Training datasets for each module.

## 5.4 Experiments

### 5.4.1 Dataset and Evaluation Metric

**Human3.6M** [52] is commonly used as the benchmark dataset for 3D human pose estimation, consisting of 3.6 millions of video frames captured in the controlled environment. It has 11 subjects, 15 kinds of action sequences and 1.5 million training images with accurate 3D annotations. Similar to [61], we use MoSH to process the marker data in the original dataset, and obtain the ground truth SMPL parameters to generate the groundtruth for $UV_l$. For a fair comparison, we use 300K data in S1, S5, S6, S7, S8 for network training, and test in S9, S11.

**3DOH** [210] utilize multi-view SMPLify-X [116] to get the 3D ground truth. The dataset is designed to have object occlusion for subjects. It contains 50,310 training images and 1,290 test images. It provides 2D, 3D annotations and SMPL parameters to generate meshes. We use the test set for evaluation purposes and the training set to train the *UVI* module.

**LSP** [56] dataset is a 2D human pose estimation benchmark. In our work, we use the [77] SMPL parameter to render the $M_i$ to train *DMP* module.

**MPI-INF-3DHP** [101] is a dataset captured with a multi-view setup mostly in indoor environments. No markers are used for the capture, so 3D pose data tend to be less accurate compared to other datasets. We use the provided training set (subjects S1 to S8) for training. We use the it to train *DMP* and *IK* module.

**Mocap dataset** We use [95] AMASS, AIST++ [82] and SPIN [68] dataset to train our occlusion aware *GIKNet*.

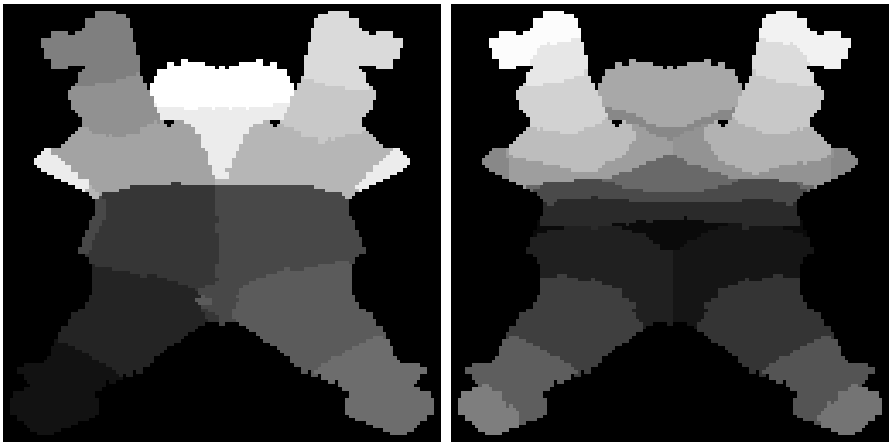


Figure 5.6: Different part segmentation choice in UV space. (Best viewed in Color)

| | H36M | |
|---|---|---|
| Method | MPJPE | MPJPE-PA |
| HMR [61] | 88.0 | 56.8 |
| DaNet [207] | 61.5 | 48.6 |
| HoloPose [38] | 60.3 | 46.5 |
| SPIN [68] | 62.5 | 41.1 |
| I2L [108] | <u>55.7</u> | 41.1 |
| DetNet [222] | 64.8 | 50.3 |
| PHMR [81] | - | 41.2 |
| DecoMR [205] | 60.5 | <u>39.3</u> |
| PyMaf [208] | 57.7 | 40.5 |
| Ours *DMP*-14 | 69.7 | 51.7 |
| Ours *IK*-14 | 67.3 | 50.6 |
| Ours *UVI*-14 | **54.7** | **38.4** |

Table 5.2: Reconstruction errors on Human3.6M dataset.

**Evaluation** We evaluate our method on H36M [52] dataset and 3DOH [210] datasets. We report Procrustes-aligned mean per joint position error (MPJPE-PA) and mean per joint position error (MPJPE) in mm. For 3DOH we also report mean per vertex error (MPVE) in mm.

### 5.4.2 Ablation Study

**14 joints vs 24 joints setting** Another way to get 24 joints prediction from *DMP* is to have a 24 joints segmentation $A_{uv}$ in UV space following SMPL setting. As shown in Fig 5.6

| | 3DOH | | |
|---|---|---|---|
| Method | MPJPE | MPJPE-PA | MPVE |
| SMPLify-X | - | 156.4 | 177.3 |
| OOH [206] | - | 58.5 | **63.3** |
| SPIN [68] | 104.3 | 68.3 | 113.4 |
| PyMAF[208] | 96.2 | - | 107.3 |
| HMR-EFT$^\star$ [57] | 75.2 | 53.1 | - |
| PARE$^\star$ [65] | <u>63.3</u> | **44.3** | - |
| Ours *DMP*-14 | 128.4 | 109.8 | - |
| Ours *IK*-14 | 112.9 | 80.8 | 133.5 |
| Ours *UVI*-14 | **58.3** | <u>44.6</u> | <u>72.3</u> |

Table 5.3: Comparison with SOTA performance on 3DOH dataset. $\star$ denotes the model trained on better ground truth data from EFT [57].

we define 14 joints setting and 24 joints setting. We run *DMP*-24 and *DMP*-14 and evaluate on the predicted $J_{initial}$. We observe the error of *DMP*-24 is much higher than *DMP*-14 as in table 5.4. The main reason is that over-segment of body parts may distribute less visible pixels to certain parts (feet, hand) and will lead to higher error.

**Occlusion vs Non-occlusion** When computing the MPJPE for $J_{initial}$, the results for visible parts (part with any pixel belong to them visible) and invisible parts differ a lot. We compare the *DMP-14* and *DMP-14-Nonoccluded* in table 5.4. We find visible parts with 87.3 mm MPJPE while the MPJPE counting invisible parts yield 128.4 mm. It tells us if the joints are visible, our *DMP* can predict relative good initial results. Thus, synthetic occlusion helps for our *DMP* module. When we remove the data augmentation techniques like synthetic occlusion [154], *DMP-14* increase to 135.4 mm.

***GIK-Net* data augmentations** We also try to remove the gaussian noise or random mask out joints data augmentation techniques for MOCAP data, which serve as input for the *GIK-Net*, to see how is the MPJPE varying. As shown in table 5.4, *IK-14 w/o gaussian noise* and *IK-14-w/o random zero* yield larger error (2.8 mm and 3.9 mm ) compared with *IK-14*. It demonstrate these data augmentation makes the *GIK-Net* more robust to noise


Figure 5.7: Pose and shape prediction from *DMP* module, *IK* module and *UVI* module. (Best viewed in Color)

.

and helps generalize to real data input.

***UVI* ablations** As the magnitude of our $UV_d$ should be symmetric, we introduce the magnitude error for $UV_d$ and its flip version. We run a model without this $\ell_{dismag}$ and observe there is 4.5 mm error increase in MPJPE metric. This is shown in table 5.4.

| | 3DOH | | |
|---|---|---|---|
| Method | MPVE | MPJPE | PMPJPE |
| *DMP*-24 | - | 246.4 | 208.5 |
| *DMP*-14 w/o synthetic occlusion | - | 135.4 | 115.7 |
| *DMP*-14-Nonoccluded | - | 87.3 | 64.7 |
| *DMP*-14 | - | 128.4 | 109.8 |
| *IK*-14 w/o gaussian noise | 138.2 | 115.7 | 82.8 |
| *IK*-14 w/o random zero | 139.5 | 116.8 | 83.2 |
| *IK*-14 | 133.5 | 112.9 | 80.8 |
| *UVI*-14 w/o *IK*-14 | 82.9 | 69.4 | 58.1 |
| *UVI*-14 w/o *DMP*-14 | 80.1 | 67.8 | 55.1 |
| *UVI*-14 w/o $\ell_{dismag}$ | 75.5 | 63.8 | 47.3 |
| *UVI*-14 | 72.3 | 58.3 | 44.6 |

Table 5.4: Ablation study about reconstruction errors on 3DOH test set. 14 and 24 denotes the number of joints setting for training and evaluations. Nonoccluded denotes when we calculate error we are not counting the part without any visible image evidence.

**Each stage performance** *DMP* module is a nonparametric method, while *IK* module is a model-based method which refines the output of the *DMP* model. *UVI* module relies on both nonparametric output and model-based output to predict the final body joint and mesh. Based on table 5.4, *DMP-14* estimate from raw images and gives inferior performance. *IK*-14 corrects the output from *DMP-14* and reduce the error by 15.5 mm. *UVI*-14 relies on both *IK-14* and *DMP-14* and further reduce MPJPE to 58.3 mm. However, if any of the previous stage output is missing, MPJPE increase by 11.1 mm (w/o *IK-14*) or 8.5 mm (w/o *DMP-14*).

### 5.4.3 Qualitative Results

We present qualitative results in Fig 5.7 including the joints prediction from *DMP*, *IK*, *UVI* modules and mesh prediction from *IK*, *UVI* modules.

**Limitations** We also show failure cases in Fig 5.8. Typical failure cases can be attributed to challenging poses (a,b,d), and crowded scenarios (c).


Figure 5.8: Failure cases. (Best viewed in color)

## 5.5 Conclusion

We propose a framework that combine the best of both worlds (nonparametric and SMPL model-based method). It predicts the initial 3D body pose from the *DMP* module, refine the predicted pose and repose the template SMPL meshes using *IK* module. Based on the nonparametric prediction from *DMP* module and model-based prediction from *IK* module, the *UVI* module inpaints and refines the prediction. To alleviate the intrinsic error introduced by joint regressor (fitting), we regress joint ($UV_j$) and mesh ($UV_l$) separately in different maps in UV space. We also introduce the magnitude loss $\ell_{dismag}$ to encourage predictions of symmetric body shape ($UV_d$). Our framework achieves state-of-the-art performance among 3D mesh-based methods on several public benchmarks. Future work can focus on extending the framework to the reconstruction of full body surfaces including hands and faces.

# Chapter 6

# Conclusions and Future Directions

## 6.1 Our Contributions

This dissertation focuses on techniques for estimation of 3D human pose from a single view, especially under occlusion and cross-dataset domain shift.

In chapter 2 we discuss how existing 3D human pose datasets have been collected and curated. In addition, we also discuss the design of networks and representations that incorporate general priors to handle 3D human pose estimation in deep learning era.

Based on the survey we do in chapter 2, in chapter 3 we carry out a systematic study of the diversity and biases present in specific datasets and its effect on cross-dataset generalization across a compendium of 5 pose datasets. We specifically focus on systematic differences in the distribution of camera viewpoints relative to a body-centered coordinate frame. Based on this observation, we propose an auxiliary task of predicting the camera viewpoint in addition to pose. Our model shows significantly improved cross-dataset generalization.

To fill in the blank that the existing datasets have no scene geometry groundtruth, then

in chapter 4 we explore the hypothesis that strong prior information about scene geometry can be used to improve pose estimation accuracy. We assemble *Geometric Pose Affordance* dataset, consisting of multi-view imagery of people interacting with a variety of rich 3D environments. We utilized a commercial motion capture system to collect gold-standard estimates of pose and construct accurate geometric 3D models of the scene geometry. To inject prior knowledge of scene constraints into existing frameworks for pose estimation from images, we introduce a view-based representation of scene geometry, a *multi-layer depth map*, which employs multi-hit ray tracing to concisely encode multiple surface entry and exit points along each camera view ray direction. We propose two different mechanisms for integrating multi-layer depth information into pose estimation: input as encoded ray features used in lifting 2D pose to full 3D, and secondly as a differentiable loss that encourages learned models to favor geometrically consistent pose estimates. We show experimentally that these techniques can improve the accuracy of 3D pose estimates, particularly in the presence of occlusion and complex scene geometry.

Finally, in chapter 5, to explore better human geometric model we propose a framework of three consecutive modules. A dense map prediction module explicitly establishes the dense UV correspondence between the image evidence and each part of the body model. The inverse kinematics module refines the key point prediction and generates a posed template mesh. Finally, a UV inpainting module relies on the corresponding feature, prediction and the posed template, and completes the predictions of occluded body shape. Our framework leverages the best of non-parametric and model-based methods and is also robust to partial occlusion. Experiments demonstrate that our framework outperforms existing 3D human estimation methods on multiple public benchmarks.

## 6.2 Limitations

In chapter 4, our dataset has limited background (mainly green background), this may lead to inferior performance if trained on and tested on in the wild images. Additionally, it would be better if we can fit a SMPL-X [116] model and together respect the scene geometry.

In chapter 5, as we use the renderer that is based on scaling orthographic camera, the generated IUV images may not align well with the silhouette of original images. It would be better if we have a differentiable renderer that not only provides projective camera but also rendered all the graphic property like surface normal, lighting and materials so that we can train.

## 6.3 Future Directions

Apart from the progress discussed above, there are still many interesting topics that we want to explore in future work.

**Joint annotations uncertainty** Different datasets are annotated using different systems or algorithms. Thus, the number of joints and each joint definition may vary. How to calibrate the variance in joint definition across datasets remain unsolved. We have seen similar work in human challenges [36] and 2d dense pose [110].

**Data augmentation** We can observe that the best video based methods on H36M model the self-occlusion in 3D human pose [18, 19], while the H36M challenges winning solution heavily really on synthetic occlusion [154] to make the model more robust to object-occlusion. How to generate more realistic or scene-aware occlusion will be a future direction which may further boost the accuracy.

**Efficient training** Most state-of-the-art algorithms [214, 151] train on 10fps-version Human36M dataset, which has 300k images. Even though it is already sub-sampled, they still need to be trained on 4 1080TI for 2 days as in [107]. Too many redundant images will make the iteration of algorithm slow and this task GPU-consuming. We thus train using different number of images (from H36M, GPA, 3DHP, 3DPW) and test on its own test set. We visualize the results in Fig 6.1. Half of the 3DHP training images are able to saturate the performance. We observe similar findings on H36M, GPA, which saturates performance with number of images around $e^{10} = (22,000)$ to $e^{11} = (60,000)$. We have not observe 3DPW with performance saturation as number of training images has not reached $e^{10} = (22,000)$. There are certainly better ways to select important samples to make the size of training set small and error on test size improved. People may select training samples based on viewpoint, key points distribution (kmeans, T-sne, CMAP as in Fig 2.4a), or active-learning.


Figure 6.1: MPJPE with different number of training images. Number of images is in log scale.

**Cross-dataset evaluation** Model trained on one dataset cannot generalize well to the other dataset. As systematically studied in section 3. We illustrate how a model performs

on the other datasets if trained on a specific dataset as in Table 6.1, and also list the root prediction (how far it is between root joint and camera center.) on the same setting as in Table 6.2. We follow the same experiments setting as [107]. This is a pretty new setting. Even though evaluated by several previous work [80, 174], there is still a big bias to be solved.

| | MPJPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|
| Te \Tr | H36M | GPA | SURREAL | 3DPW | 3DHP |
| H36M | **53.2** | 110.5 | 107.1 | 125.1 | 108.4 |
| GPA | 105.2 | **53.9** | 86.8 | 111.7 | 90.5 |
| SURREAL | 118.6 | 103.2 | **37.2** | 120.8 | 108.2 |
| 3DPW | 108.7 | 116.4 | 114.2 | **100.6** | 113.3 |
| 3DHP | 111.8 | 123.9 | 120.3 | 139.7 | **91.9** |

Table 6.1: Cross-dataset evaluation based on [107]. Te stands for testing set and Tr stands for training set. Table credit [182]

| | MRPE (in mm, lower is better) | | | | |
|---|---|---|---|---|---|
| Te \Tr | H36M | GPA | SURREAL | 3DPW | 3DHP |
| H36M | **132.3** | 429.8 | 334.1 | 1214.8 | 1041.9 |
| GPA | 588.0 | **142.5** | **308.6** | 1003.1 | 744.8 |
| SURREAL | 1664.2 | 1153.5 | 119.1 | 1619.8 | 2227.2 |
| 3DPW | 526.4 | 497.1 | 410.8 | **615.5** | 738.8 |
| 3DHP | 524.7 | 417.6 | 411.9 | 810.0 | **288.6** |

Table 6.2: RootNet [107] cross-dataset evaluation (MRPE, unit in mm). Te stands for testing set and Tr stands for training set.

**Robust testing** Even though we [182] reduce the model bias on cross-dataset setting by 4 mm per dataset, a large gap remains. How do we correctly evaluate our model with less bias? We make a MIX test set to test these model ability as in Fig 6.2. For GPA, We select part of occlusion + close2geometry set, which is hard in general. For H36M we uniformly sample 1/64 of original test set (this is the same with some of the chapter testset). For SURREAL/3DHP we use the original test set. For 3DPW we uniformly sample 1/4 of original testset. In total our MIX test set has 29,699 samples. The MPJPE of each model is shown in 6.2. We expect more researchers working on this direction to reduce the cross dataset evaluation bias.

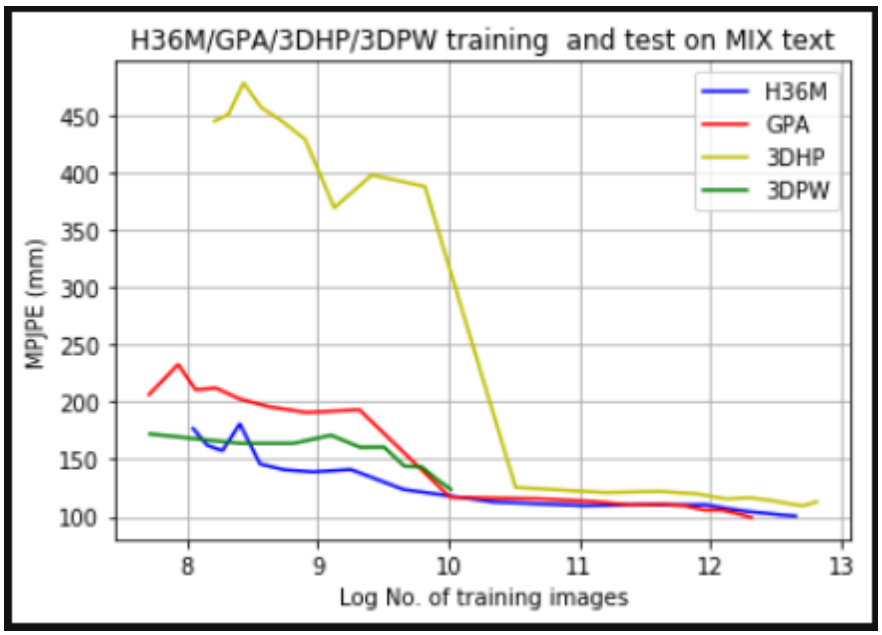



Figure 6.2: MPJPE with different number of training images while evaluating on the same MIX test set. Number of images is in log scale.

# Bibliography


[1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 2015.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[3] A. Arnab, C. Doersch, and A. Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPR*, 2019.

[4] A. Balestrino, G. D. Maria, and L. Sciavicco. Robust control of robotic manipulators. In *IFAC Proceedings Volumes*, 1984.

[5] L. Bertoni, S. Kreiss, and A. Alahi. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation. In *ICCV*, 2019.

[6] 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 *ECCV*, 2016.

[7] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[8] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, 2019.

[9] Z. Cao, H. Gao, K. Mangalam, Q. Cai, M. Vo, and J. Malik. Long-term human motion prediction with scene context. In *ECCV*, 2020.

[10] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[11] K.-C. Chan, C.-K. Koh, and C. S. G. Lee. A 3d-point-cloud feature for human-pose estimation. In *ICRA*, 2013.

[12] J. Charles, T. Pfister, M. Everingham, and A. Zisserman. Automatic and efficient human pose estimation for sign language videos. In *IJCV*, 2013.

[13] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *CVPR*, 2017.

[14] C.-H. Chen, A. Tyagi, A. Agrawal, D. Drover, R. MV, S. Stojanov, and J. M. Rehg. Unsupervised 3d pose estimation with geometric self supervision. In *CVPR*, 2019.

[15] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *NeurIPS*, 2016.

[16] X. Chen, K.-Y. Lin, W. Liu, C. Qian, X. Wang, and L. Lin. Weakly-supervised discovery of geometry-aware representationfor 3d human pose estimation. In *CVPR*, 2019.

[17] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan, and X. Xie. Nonparametric structure regularization machine for 2d hand pose estimation. In *WACV*, 2020.

[18] Y. Cheng, B. Yang, B. Wang, and R. T.Tan. 3d human pose estimation using spatio-temporal networks with explicit occlusion training. In *AAAI*, 2020.

[19] Y. Cheng, B. Yang, B. Wang, W. Yan, and R. T. Tan. Occlusion-aware networks for 3d human pose estimation in video. In *ICCV*, 2019.

[20] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

[21] H. Ci, C. Wang, X. Ma, and Y. Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, 2019.

[22] H. M. Clever, Z. Erickson, A. Kapusta, G. Turk, C. K. Liu, and C. C. Kemp. Bodies at rest: 3d human pose and shape estimation from a pressure image using synthetic data. In *CVPR*, 2020.

[23] R. Dabral, A. Mundhada, U. Kusupati, S. Afaque, A. Sharma, and A. Jain. Learning 3d human pose from structure and motion. In *ECCV*, 2018.

[24] R. Díaz, M. Lee, J. Schubert, and C. C. Fowlkes. Lifting gis maps into strong geometric context for scene understanding. In *WACV*, 2016.

[25] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

[26] M. Fabbri, F. Lanzi, S. Calderara, S. Alletto, and R. Cucchiara. Compressed volumetric heatmaps for multi-person 3d pose estimation. In *CVPR*, 2020.

[27] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 2018.

[28] H.-S. Fang, G. Lu, X. Fang, J. Xie, Y.-W. Tai, and C. Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. In *CVPR*, 2018.

[29] 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 *AAAI*, 2018.

[30] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In *Communications of the ACM*. ACM, 1981.

[31] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012.

[32] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes. Parsing occluded people. In *CVPR*, 2014.

[33] J. Gibson. The ecological approach to visual perception. In *Boston: Houghton Mifflin*, 1979.

[34] M. Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. In *SIGGRAPH*, 1985.

[35] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NeurIPS*, 2020.

[36] K. Gong, Y. Gao, X. Liang, X. Shen, M. Wang, and L. Lin. Graphonomy: Universal human parsing via graph transfer learning. In *CVPR*, 2019.

[37] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *CVPR*, 2018.

[38] R. A. Guler and I. Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *CVPR*, 2019.

[39] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011.

[40] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *CVPR*, 2019.

[41] K. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *CVPR*, 2019.

[42] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *ECCV*, 2016.

[43] N. Hasler, B. Rosenhahn, T. Thormahle, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, 2009.

[44] M. Hassan, V. Choutas, D. Tzionas, and M. J. Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, 2019.

[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *CVPR*, 2017.

[46] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[47] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, 2006.

[48] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[49] M. R. I. Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *ECCV*, 2018.

[50] M. R. I. Hossain and J. J. Little. Exploiting temporal information for 3d pose estimation. In *ECCV*, 2018.

[51] Y. Huang, F. Bogo, C. Lassner, A. Kanazawa, P. V. Gehler, J. Romero, I. Akhter, and M. J. Black. Towards accurate marker-less human shape and pose estimation over time. In *3DV*, 2017.

[52] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. In *PAMI*, 2014.

[53] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *ECCV*, 2018.

[54] U. Iqbal, P. Molchanov, and J. Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *CVPR*, 2020.

[55] H. Jiang and K. Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *CVPR*, 2017.

[56] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.

[57] H. Joo, N. Neverova, and A. Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-thewild 3d human pose estimation. In *Arxiv*, 2020.

[58] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social interaction capture. In *PAMI*, 2017.

[59] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands and bodies. In *CVPR*, 2016.

[60] H. Jung, Y. Suh, G. Moon, and K. M. Lee. A sequential approach to 3d human pose estimation: Separation of localization and identification of body joints. In *ECCV*, 2016.

[61] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[62] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik. Learning 3d human dynamics from video. In *CVPR*, 2019.

[63] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[64] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[65] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black. Pare: Part attention regressor for 3d human body estimation. In *ICCV*, 2021.

[66] M. Kocabas, S. Karagoz, and E. Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *CVPR*, 2019.

[67] I. Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid- and high-level vision using diverse datasets and limited memory. In *Arxiv*, 2016.

[68] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[69] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019.

[70] D. Kong, Y. Chen, H. Ma, X. Yan, and X. Xie. Adaptive graphical model network for 2d handpose estimation. In *BMVC*, 2019.

[71] D. Kong, H. Ma, Y. Chen, and X. Xie. Rotation-invariant mixed graphical model network for 2d hand pose estimation. In *WACV*, 2020.

[72] D. Kong, H. Ma, and X. Xie. Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. In *arXiv*, 2020.

[73] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *RSS*, 2013.

[74] P. Krahenbuhl. Free supervision from video games. In *CVPR*, 2018.

[75] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty. Self-supervised 3d human pose estimation via part guided novel image synthesis. In *CVPR*, 2020.

[76] K. Lasinger, R. Ranftl, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. In *Arxiv*, 2019.

[77] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people – closing the loop between 3d and 2d human representations. In *CVPR*, 2017.

[78] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015.

[79] K. Lee, I. Lee, and S. Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *ECCV*, 2018.

[80] C. Li and G. H. Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *CVPR*, 2019.

[81] 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 *CVPR*, 2021.

[82] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021.

[83] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014.

[84] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *ICCV*, 2015.

[85] X. Li, S. Liu, K. Kim, X. Wang, M.-H. Yang, and J. Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *CVPR*, 2019.

[86] Z. Li, X. Wang, F. Wang, and P. Jiang. On boosting single-frame 3d human pose estimation via monocular videos. In *ICCV*, 2019.

[87] J. Lin and G. H. Lee. Trajectory space factorization for deep video-based 3d human pose estimation. In *BMVC*, 2019.

[88] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021.

[89] K. Lin, L. Wang, and Z. Liu. Mesh graphormer. In *ICCV*, 2021.

[90] M. Lin, L. Lin, X. Liang, K. Wang, and H. Cheng. Recurrent 3d pose sequence machines. In *CVPR*, 2017.

[91] T.-Y. Lin, a. S. B. Michael Maire, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[92] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *SIGGRAPH*, 2015.

[93] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.

[94] 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 *BMVC*, 2021.

[95] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019.

[96] S. Maji, L. Bourdev, and J. Malik. Action recognition using a distributed representation of pose and appearance,. In *CVPR*, 2011.

[97] E. Marinoiu, M. Zanfir, V. Olaru, and C. Sminchisescu. 3d human sensing, action and emotion recognition in robot assisted therapy of children with autism. In *CVPR*, 2018.

[98] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

[99] K. Matzen and N. Snavely. Nyc3dcars: A dataset of 3d vehicles in geographic context. In *ICCV*, 2013.

[100] L. McInnes, J. Healy, N. Saul, and L. Grossberger. Umap: Uniform manifold approximation and projection. In *The booktitle of Open Source Software*, 2018.

[101] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.

[102] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. In *SIGGRAPH*, 2020.

[103] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3DV*, 2018.

[104] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. In *ToG*, 2017.

[105] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and G. andreas. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[106] A. Monszpart, P. Guerrero, D. Ceylan, E. Yumer, and N. J. Mitra. imapper: Interaction-guided joint scene and human motion mapping from monocular videos. In *Arxiv*, 2018.

[107] G. Moon, J. Chang, and K. M. Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019.

[108] G. Moon and K. M. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020.

[109] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017.

[110] N. Neverova, D. Novotny, and A. Vedaldi. Correlated uncertainty for learning dense correspondences from noisy labels. In *NeurIPS*, 2019.

[111] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[112] A. Nibali, Z. He, S. Morgan, and L. Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *WACV*, 2019.

[113] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *ICCV*, 2019.

[114] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, 2018.

[115] 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 *ECCV*, 2018.

[116] 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 *CVPR*, 2019.

[117] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, 2017.

[118] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017.

[119] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *CVPR*, 2017.

[120] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, 2018.

[121] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019.

[122] D. Pavllo, D. Grangier, and M. Auli. Quaternet: A quaternion-based recurrent model for human motion. In *BMVC*, 2018.

[123] A. Pirinen, E. Gärtner, and C. Sminchisescu. Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction. In *NeurIPS*, 2019.

[124] G. Pons-Moll and B. Rosenhahn. Model-based pose estimation. In *Visual Analysis of Humans*, 2011.

[125] A.-I. Popa, M. Zanfir, and C. Sminchisescu. Deep multitask architecture for integrated 2d and 3d human sensing. In *CVPR*, 2017.

[126] A. Qammaz and A. A. Argyros. Mocapnet: Ensemble of snn encoders for 3d human pose estimation in rgb images. In *BMVC*, 2019.

[127] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng. Cross view fusion for 3d human pose estimation. In *ICCV*, 2019.

[128] U. Raf, J. Gall, and B. Leibe. A semantic occlusion model for human pose estimation from a single depth image. In *CVPRW*, 2015.

[129] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation and gender recognition. In *TPAMI*, 2016.

[130] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In *CVPR*, 2020.

[131] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[132] H. Rhodin, V. Constantin, I. Katircioglu, M. Salzmann, and P. Fua. Neural scene decomposition for multi-person motion capture. In *CVPR*, 2019.

[133] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *ECCV*, 2018.

[134] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua. Learning monocular 3d human pose estimation from multi-view images. In *CVPR*, 2018.

[135] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, 2018.

[136] C. Rockwell and D. Fouhey. Full-body awareness from partial observations. In *ECCV*, 2020.

[137] G. Rogez and C. Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *NeurIPS*, 2016.

[138] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net: Localization-classification-regression for human pose. In *CVPR*, 2017.

[139] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. In *PAMI*, 2019.

[140] C. Rother, V. Kolmogorov, and A. Blake. "grabcut" interactive foreground extraction using iterated graph cuts. In *ToG*, 2004.

[141] N. Saini, E. Price, R. Tallamraju, R. Enficiaud, R. Ludwig, I. Martinović, A. Ahmad, and M. Black. Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In *ICCV*, 2019.

[142] S. Sharma, P. T. Varigonda, P. Bindal, A. Sharma, and A. Jain. Monocular 3d human pose estimation by generation and ordinal ranking. In *ICCV*, 2019.

[143] D. Shin, C. Fowlkes, and D. Hoiem. Pixels, voxels and views: A study of shape representations for single view 3d object shape prediction. In *CVPR*, 2018.

[144] D. Shin, Z. Ren, E. Sudderth, and C. Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *ICCV*, 2019.

[145] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronizedvideo and motion capture dataset and baseline algorithm forevaluation of articulated human motion. In *IJCV*, 2010.

[146] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[147] S. Song, F. Yu, A. Zeng, A. X. Chang, M. lis Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[148] S. Spurlock and R. Souvenir. Multimodal 3d human pose estimation from a single image. In *3DV*, 2019.

[149] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[150] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *ICCV*, 2017.

[151] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *ECCV*, 2018.

[152] Y. Sun, Y. Ye, W. Liu, W. Gao, Y. Fu, and T. Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *ICCV*, 2019.

[153] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. How robust is 3d human pose estimation to occlusion? In *Arxiv*, 2018.

[154] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe. Synthetic occlusion augmentation with volumetric heatmaps for the 2018 eccv posetrack challenge on 3d human pose estimation. In *Arxiv*, 2018.

[155] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020.

[156] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017.

[157] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019.

[158] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. In *BMVC*, 2016.

[159] B. Tekin, A. Rozantsev, V. Lepetit, and P. Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016.

[160] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *CVPR*, 2017.

[161] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015.

[162] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[163] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[164] M. Trumble, G. andrew, C. Malleson, A. Hilton, and J. Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*, 2017.

[165] M. Trumble, A. Gilbert, A. Hilton, and J. Collomosse. Deep autoencoder for combined human pose estimation and body model upscaling. In *ECCV*, 2018.

[166] S. Tulsiani, S. Gupta, D. Fouhey, A. AEfros, and J. Malik. Factoring shape, pose and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[167] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.

[168] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *ICCV*, 2017.

[169] H.-Y. F. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki. Self-supervised learning of motion capture. In *CVPR*, 2009.

[170] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *CVPR*, 2019.

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

[172] R. Villegas, J. Yang, D. Ceylan, and H. Lee. Neural kinematic networks for unsupervised motion retargetting. In *CVPR*, 2018.

[173] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018.

[174] B. Wandt and B. Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *CVPR*, 2019.

[175] C. Wang, Y. Wang, Z. Lin, and A. L. Yuille. Robust 3d human pose estimation from single images or video sequences. In *PAMI*, 2018.

[176] S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015.

[177] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, 2019.

[178] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *CVPR*, 2017.

[179] Z. Wang, H. Chen, X. Li, C. Liu, Y. Xiong, J. Tighe, and C. Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *WACV*, 2022.

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

[181] Z. Wang, X. Liu, L. Chen, L. Wang, Y. Qiao, X. Xie, and C. Fowlkes. Structured triplet learning with pos-tag guided attention for visual question answering. In *WACV*, 2018.

[182] Z. Wang, D. Shin, and C. Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *ECCV 3DPW workshop*, 2020.

[183] Z. Wang, L. Wang, W. Du, and Y. Qiao. Exploring fisher vector and deep networks for action spotting. In *CVPRW*, 2015.

[184] Z. Wang, L. Wang, Y. Wang, B. Zhang, and Y. Qiao. Weakly supervised patchnets: Describing and aggregating local patches for scene recognition. In *TIP*, 2017.

[185] Z. Wang, Y. Wang, L. Wang, and Y. Qiao. Codebook enhancement of vlad representation for visual recognition. In *ICASSP*, 2016.

[186] P. Weinzaepfel, R. Brégier, H. Combaluzier, V. Leroy, and G. Rogez. DOPE: Distillation Of Part Experts for whole-body 3D pose estimation in the wild. In *ECCV*, 2020.

[187] W. A. Wolovich and H. Elliott. A computational technique for inverse kinematics. In *CDC*, 1984.

[188] H. Wu and B. Xiao. 3d human pose estimation via explicit compositional depth maps. In *AAAI*, 2020.

[189] D. Xiang, H. Joo, and Y. Sheikh. Monocular total capture: Posing face, body and hands in the wild. In *CVPR*, 2019.

[190] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[191] R. Xie, C. Wang, and Y. Wang. Metafuse: A pre-trained fusion model for human pose estimation. In *CVPR*, 2020.

[192] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[193] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *ICCV*, 2019.

[194] Y. Xu, S.-C. Zhu, and T. Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019.

[195] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NeurIPS*, 2016.

[196] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, 2018.

[197] Z. Yang, W. Zhu, W. Wu, C. Qian, Q. Zhou, B. Zhou, and C. C. Loy. Transmomo: Invariance-driven unsupervised video motion retargeting. In *CVPR*, 2020.

[198] Y. Yao, Y. Jafarian, and H. S. Park. Monet: Multiview semi-supervised keypoint detection via epipolar divergence. In *ICCV*, 2019.

[199] R. A. Yeh, Y.-T. Hu, and A. G. Schwing. Chirality nets for human pose regression. In *NeurlPS*, 2019.

[200] X. Yin and X. Liu. Multi-task convolutional neural network for pose-invariant face recognition. In *TIP*, 2017.

[201] S. Yu, Y. Yun, L. Wu, G. Wenpeng, F. YiLi, and M. Tao. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *ICCV*, 2019.

[202] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park. Humbi: A large multiview dataset of human body expressions. In *CVPR*, 2020.

[203] A. Zanfir, E. Marinoiu, and C. Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes. In *CVPR*, 2018.

[204] A. Zanfir, E. Marinoiu, M. Zanfir, A.-I. Popa, and C. Sminchisescu. Deep network for the integrated 3d sensing of multiple people in natural images. In *NeurIPS*, 2018.

[205] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang. 3d human mesh regression with dense correspondence. In *CVPR*, 2020.

[206] T. Zhan, B. Huang, and Y. Wangu. Object-occluded human shape and pose estimation from a single color image. In *CVPR*, 2020.

[207] H. Zhang, J. Cao, G. Lu, W. Ouyang, and Z. Sun. Learning 3d human shape and pose from dense body parts. In *PAMI*, 2020.

[208] H. Zhang, Y. Tian, X. Zhou, W. Ouyang, Y. Liu, L. Wang, and Z. Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021.

[209] J. Y. Zhang, S. Pepose, H. Joo, D. Ramanan, J. Malik, and A. Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020.

[210] T. Zhang, B. Huang, and Y. Wang. Object-occluded human shape and pose estimation from a single color image. In *CVPR*, 2020.

[211] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu. Learning to reconstruct shapes from unseen classes. In *NeurIPS*, 2018.

[212] Z. Zhang, C. Wang, W. Qin, and W. Zeng. Fusing wearable imus with multi-view images for human pose estimation: A geometric approach. In *CVPR*, 2020.

[213] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, 2019.

[214] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *ICCV*, 2019.

[215] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *ICCV*, 2017.

[216] X. Zhou, A. Karpur, L. Luo, and Q. Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *ECCV*, 2018.

[217] X. Zhou, S. Liu, G. Pavlakos, V. Kumar, and K. Daniilidis. Human motion capture using a drone. In *ICRA*, 2018.

[218] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. In *ECCVW*, 2016.

[219] X. Zhou, M. Zhu, S. Leonardos, K. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *CVPR*, 2016.

[220] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. In *PAMI*, 2018.

[221] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.

[222] Y. Zhou, M. Habermann, I. Habibie, A. Tewari, C. Theobalt, and F. Xu. Monocular real-time full body capture with inter-part correlations. In *CVPR*, 2021.

[223] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, and F. Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *CVPR*, 2020.

[224] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.

[225] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019.