Towards Multi-Person 3D Pose Estimation in Natural Videos

Renshu Gu

A dissertation

submitted in partial fulfillment of the

requirements for the degree of:

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Jenq-Neng Hwang (Chair)

Blake Hannaford

De Meng

Program Authorized to Offer Degree:

Electrical and Computer Engineering

University of Washington

Abstract

Renshu Gu

Chair of the Supervisory Committee:

Jenq-Neng Hwang

Electrical and Computer Engineering

Despite the increasing need of analyzing human poses on the street and in the wild, multi-person 3D pose estimation using static or moving monocular camera in real-world scenarios remains a challenge, requiring large-scale training data or high computation complexity due to the high degrees of freedom in 3D human poses.

To address these challenges, a novel scheme, Hierarchical 3D Human Pose Estimation (H3DHPE), is proposed to effectively track and hierarchically estimate 3D human poses in natural videos in an efficient fashion. Torso estimation is formulated as a Perspective-N-Point (PNP) problem, limb pose estimation is solved as an optimization problem, and the high dimensional pose estimation is hierarchically addressed efficiently.

As an extension to Hierarchical 3D Human Pose Estimation (H3DHPE), Universal Hierarchical 3D Human Pose Estimation (UH3DHPE) is proposed to handle the case of an occluded or inaccurate 2D torso keypoints, which makes torso-first estimation in H3DHPE unreliable. An effective method to directly estimate limb poses without building upon the estimated torso pose is proposed, and the torso pose can then be further refined to form the hierarchy in a bottom-up fashion. An adaptive merging strategy is proposed to determine the best hierarchy. The advantages of the proposed unsupervised methods are validated on various datasets including a lot of natural real-world scenes.

For better evaluation and future research, a unique dataset called Moving camera Multi-Human interactions (MMHuman) is collected, with accurate MoCap ground truth, for multi-person interaction scenarios recorded by a monocular moving camera. Superior performance is shown on the newly collected MMHuman compared to state-of-the-art methods, including supervised methods, proving that our unsupervised solution generalize better to natural videos.

To further tackle the problem of long term occlusions, a deep neutral network (DNN) solution is explored for trajectory recovery. To our best knowledge, it's the first to use temporal gated convolutions to recover missing poses and address the occlusion issues in the pose estimation. A simple yet effective approach is proposed to transform normalized poses to the global trajectory into the camera coordinate.

Contents

List of Tables

List of Figures

Acknowledgements

First I would like to express my utmost gratitude to my adviser, Prof. Jenq-Neng Hwang, for his guidance throughout my PhD study. From him I have learned a lot about how a successful scholar tackles open problems. His attitude towards research has been a role model. All of these would be my treasure for life.

I am very grateful to my committee members, Prof. Blake Hannaford, Prof. Kevin McQuade and Dr. De Meng, for the collaborations and discussions they participate in, and the valuable suggestions they have given.

I am thankful to the colleagues, alumni, and visiting scholars I have met at the Information Processing Lab (IPL), Chun-Te Chu, Shian-Ru Ke, Xiang Chen, Kuan-Hui Lee, Meng-Che Chuang, Younggun Lee, Jounsup Park, Gaoang Wang, Zheng Tang, Tsung-Wei Huang, Li Chen, Yaochung Liang, Haotian Zhang, Jiarui Cai, Yizhou Wang, Zhongyu Jiang, Chengqian Ma, Xinyu Yuan, Hung-Min Hsu, Yanting Zhang, Fangyi Zhu, Ping Zhang, Aotian Zheng, Zhichao Lei, Wenhuan Wei, Adwin Jahn, Qiuyu Chen, Xu Liu, Wei Huang, Tao Liu, Yen-Shuo Lin, Chris Ma, Mitchell Hsu, Shaoyu Wang and so on, for their help, discussions and collaborations during my PhD study.

I also thank my friends Yaxuan Zhou, Jiayun Li, Yu Jin, Xichen Jiang, Jianqing Qi, Zheng Li , Wei Niu et al., for helping me record experimental data.

I am grateful of having all the friends in Seattle and outside Seattle. You have been the lights during my life of studying abroad, and I hope it would shine through my life.

I am grateful to my dearest family: my father, mother, grandpa, grandma, my fiancé, and all the family members, for their endless love, support and encouragement. Without them, this journey would never have been possible.

This dissertation is dedicated to them.

1. INTRODUCTION

Human Pose Estimation (HPE) is central to the analysis of human behaviors in images and videos. It has tons of applications such as video analytics, gaming, health care, autonomous driving, etc. Existing 3D human pose estimation calls for large-scale training data, or high computation complexity due to the high degrees of freedom in 3D human poses. In recent years, there have been many reports on 3D human pose estimation in the experimental setting. However, there is still lack of analysis on natural videos due to the lack of 3D ground truth for natural scenes.

Deep learning methods that use powerful training from 3D motion capture (MoCap) data offer a popular solution to 3D human pose estimation [18,19,28,30]. Yet, a major challenge in 3D human pose estimation is that 3D supervision is limited in quantity and challenging to obtain at a large scale. 3D MoCap training data are typically acquired in controlled environments with limited variations. First, it may not generalize well to real-world data outside the training set. Most recent studies like [67] already verifies that even if training-based method achieves very accurate result on existing datasets, they do not generalize to the complexity of the real world. Second, obtaining 3D training data is expensive. The lack of large-scale training data and the lack of variation becomes a bottleneck. Third, existing methods might over-fit to sparse camera settings and bear poor generalization capabilities. Even with adequate training data, it is unclear how the space of 3D poses can be uniformly sampled. Moreover, many training methods focus on single person 3D pose estimation with the subject being at the center of the image, which makes it easy to associate estimated single-person body joints along the time for subsequent action and behavior analyses. On the other hand, handling multi-person 3D pose estimation where people can be interacting and occluding one another is harder. To achieve that using many existing methods, it requires detecting and cropping the person, and sometimes the unreliable detection and cropping of the image adversely affect the performance in real-world images or videos. Also, temporal information is still not well exploited in many of the existing works.

In this dissertation, with the development of research, a new video dataset called the Moving camera Multi-person Human interaction (MMHuman) dataset is collected for multi-person interactions by monocular moving cameras, under a MoCap ground truth data acquisition environment. The contribution of collecting this dataset is two folds: First, it provides accurate ground truth data for evaluation on natural human interaction scenes. Second, it serves as complementary data with 3D ground truth available in the literature.

To address the challenge of multi-person 3D pose estimation in natural videos using static or moving monocular camera, an efficient unsupervised Hierarchical 3D Human Pose Estimation (H3DHPE) method is proposed. The proposed H3DHPE allows to efficiently reconstruct 3D human poses for multiple people in monocular image sequences with arbitrary camera motion without training on 3D ground truth. Unlike

existing methods that feature high degrees of freedom (DoFs) in pose space, the pose space is structured in a hierarchical fashion to tackle the problem efficiently. H3DHPE first utilizes recent advances in 2D pose estimation, for example, OpenPose [1], tracks multiple people to use temporal information, and then estimating each person's 3D human poses hierarchically with body geometric constraints. When estimating each person's poses, a prior flexible human model that contains angle constraints and bone length constraints is applied. Instead of trying to solve all the poses in high dimensions simultaneously, the torso pose is first estimated, and then limb poses are estimated in a hierarchical fashion. Using 2D joints of multiple people from OpenPose [1] as an intermediate step, the proposed H3DHPE does not need to crop bounding boxes, and is robust to position changes. H3DHPE is demonstrated to produce smooth and natural 3D human poses on real-world datasets, such as KITTI [2], ETH [3], DALY [4] and UCLA HHOI [46] that are of high interest to many applications. H3DHPE is validated on public dataset Human3.6M [5], which has the ground truth of all joints on 15 common human actions and is widely used in 3D human pose estimation.

As an extension to Hierarchical 3D Human Pose Estimation (H3DHPE), Universal Hierarchical 3D Human Pose Estimation (UH3DHPE) is proposed in this dissertation, to address the remaining issues in H3DHPE. To handle the case of an occluded or inaccurate 2D torso keypoints, which makes torso-first estimation in H3DHPE unreliable, an effective method to directly estimate limb poses without building upon the estimated torso pose is proposed, and the torso pose can then be further refined to form the hierarchy in a bottom-up fashion. An adaptive merging strategy is proposed to determine the best hierarchy. We demonstrate the advantage of our proposed UH3DHPE over existing unsupervised solutions, by comparing the performances on UCLA HHOI [46], Human3.6M [5], UWHHI [51], and our newly collected dataset MMHuman.

To further tackle the problem of severe and/or long-term occlusions, which is unrealistic to address without exploiting temporal information, we explore a deep neutral network (DNN) solution for trajectory recovery. We propose a temporal regression network with gated convolution module to transform 2D joints to 3D and recover the missing occluded joints in the meantime. A simple yet effective localization approach is further conducted to transform the normalized pose to the global trajectory.

In summary, the major contributions of my PhD work are:

1. A pipeline that integrates visual odometry, 3D human pose estimation, and exploit the temporal information using a powerful tracking method is proposed.

2. An unsupervised 3D HPE method, H3DHPE, is proposed. H3DHPE formulates the torso estimation as a Perspective-N-Point (PNP) problem and provide a highly efficient solution. For each limb, we formulate and solve an optimization problem. In H3DHPE, an effective occlusion handling strategy is proposed utilizing keypoint confidence in case of missing or erroneous 2D human pose estimation. With the

hierarchical problem solving structure, we greatly reduce complexity in high dimensional pose space. Moreover, each limb will not interfere with one another.

3. Building upon H3DHPE, Universal 3D Hierarchical Human Pose Estimation method (UH3DHPE) is proposed in this dissertation. In this universal version, to solve the issues when torso keypoints are missing or unreliable, a bottom-up unsupervised 3D human pose estimation algorithm that is robust to unreliable or missing torso keypoints detections is proposed. UH3DHPE can overcome various intra-person or inter-person occlusion scenarios, without requiring 3D ground truth poses for training. We also propose an adaptive merging strategy to prioritize the best pose estimation hierarchy.

4. To further tackle the problem of long term occlusions, we explore a deep neutral network (DNN) solution for trajectory recovery. To our best knowledge, we are the first to use temporal gated convolutions to recover missing poses and address the occlusion issues in the pose estimation. The idea of missing pose recovery is inspired from the image inpainting tasks. A simple yet effective approach is proposed to transform normalized poses to the global trajectory into the camera coordinate.

5. A new dataset, Moving camera Multi-Human interaction (MMHuman), is presented, providing accurate MoCap ground truth for multi-person interaction scenarios recorded by a monocular moving camera.

6. Experiments are performed on a variety of natural videos, possibly containing multiple people interactions from a moving camera, which are lacking in the current literature yet critical in today's applications. The experimental results proves the advantages of the proposed methods, and shows great potential in real-world applications.

The organization of the dissertation is as follows.

Chapter 1: We introduce the background and motivations of our research.

Chapter 2: The related works are explained in detail, including related 2D human pose estimation, 3D human pose estimation and related datasets.

Chapter 3: A new Moving camera Multi-Human interaction (MMHuman) dataset is introduced.

Chapter 4: Hierarchical 3D Human Pose Estimation (H3DHPE) is presented.

Chapter 5: We propose Universal Hierarchical 3D Human Pose Estimation (UH3DHPE), which is an extended work of Hierarchical 3D Human Pose Estimation.

Chapter 6: We present our exploration of trajectory recovery using deep neural network (DNN).

Chapter 7: Conclusions and Future Work are discussed.

2. RELATED WORK

2.1. *2D Human Pose Estimation from Monocular Cameras*

As a crucial task to facilitate analyses of human actions and activities in image and videos, human pose estimation in 2D based on monocular cameras has been well studied. Recent efforts on deep learning approaches, mainly CNN based [1,6,7,8,9,10,11,12,13,14,15,68], show reliable results for multiple people. It is relatively easier to label 2D human pose without the need of experimental settings; therefore, acquiring 2D training data is less a problem. The performance on the MPII benchmark [16] has become saturated in the past three years, reaching more than 90% percentage of correct keypoints (PCKh) with a successful predicted human joint localization being within 50% of the head segment length to the ground truth joint (PCKh@0.5) [12,13,14,15].

2.2. *3D Human Pose Estimation from Monocular Cameras*

Unlike 2D human pose estimation, 3D human pose estimation from monocular cameras is far from mature. Many early approaches [17] are based on appearance models (e.g., silhouettes) and perform tracking using stochastic search with kinematic constraints. However, silhouette extraction can be unreliable due to complex backgrounds, occlusions, and moving cameras. When deep learning prospers, researchers first turn to 3D training data to solve the problem. 3D training data are obtained by the MoCap system in constrained environments. Later, to tackle the more challenging task of 3D pose estimation in the wild, some researchers find 3D training data not enough, and use 2D training data in addition. Related methods can be roughly grouped into 3 categories. The first two categories are called Direct3D methods as they need 3D MoCap training data. In contrast, the third category, which our method falls within, does not necessarily need 3D training data.

(1) One-stage methods: 3D human pose estimation methods are directly trained and tested based on multi-view 3D MoCap data and corresponding ground truth of 3D joints [18,19]. Li and Chan [18] pretrain their network with maps for 2D joint classification, and use a multi-task framework to jointly train pose regression and body part detectors. In [19] the authors propose a framework that can be interpreted as a special form of structured support vector machines where the joint feature space is discriminatively learned. An end-to-end framework Human Mesh Recovery (HMR) is proposed in [60] for reconstructing a full 3D mesh of a human body from a single RGB image.This category of methods depends heavily on 3D training data, which is lacking in the current literature. To exploit human pose data, some researchers adopt Joint estimation from

multiple sources and jointly solves 2D and 3D pose estimation [25, 26, 27, 28, 40, 29, 30, 41]. In [28], Rogez et al. propose an architecture for joint 2D and 3D human pose estimation in natural images, key to which is the generation and scoring of a number of pose proposals. In their extended work [40], multiple full-body 2D and 3D pose hypotheses are generated in different regions of the image, which are then efficiently sampled, scored and refined using an end-to-end CNN architecture inspired by the latest work on object detection, and eventually combined to estimate both the location and the 2D-3D pose of the individuals present in the observed scene. Both [28] and [40] only consider single image, and neither exploit temporal information nor handle occlusions. In [41], Luvizon et al. propose a multitask framework that can be trained with data from different categories simultaneously for joint 2D/3D pose estimation from still images and human action recognition from video sequences. Their method does not handle multiple people cases, and does not handle occlusions. A fine discretization of the 3D space around the subject is proposed in [61], which employs a coarse-to-fine prediction with a single RGB image input.

For one-stage 3D pose estimation methods, images features are directly trained with 3D ground truth. Localizing and cropping the subject greatly affects the image features, which are also subject to uncontrolled real-world variations like occlusion, lighting conditions, image quality, etc. Since there are limited data available with 3D ground truth, end-to-end training cannot easily adjust to variations of real-world scenes.

(2) Two-stage methods: the problem is tackled with two steps. **First,** detect the 2D joints by a 2D pose detector; **second,** estimate 3D pose from 2D poses. In other words, methods in this category use the 2D results as an intermediate step. Ramakrishna et al. [22] represents a 3D pose by a linear combination of a set of base poses that are learned from motion databases by minimizing the reprojection error directly in the high dimensional pose space. To overcome this high dimensional search problem, Wang et al. [23] further extend the work in [22] by enforcing the length proportions of eight selected limbs to be constant. Martinez et al. [62] show that given accurate 2D keypoints, predicting 3D poses is in fact relatively straightforward. SMPLify [20] fits a statistical body shape model to the 2D joints by minimizing an objective function that penalizes the error between the projected 3D model joints and detected 2D joints. In recent work [24], Wang et al. represent 3D poses by a sparse combination of bases which encode structural pose priors to reduce the lifting ambiguity. Their system outputs K candidate 3D poses and improve 3D pose estimates by post-processing as well as exploiting temporal cues. In [27], Zhou et al solve the correspondence between video and 3D motion capture data. To overcome ambiguities and occlusions that cannot be easily recovered using a single frame, recent works utilize temporal information to overcome the issues caused by occlusion and improve temporal consistency. A representative work, VideoPose3D [63], employs dilated temporal convolutions to capture long-term temporal information. The depth of 2D joints are predicted to obtain the 3D poses in [64]. To utilize temporal information, [65] designed a sequence-to-sequence network composed

of layer-normalized LSTM units with shortcut connections. A deep learning-based framework that utilizes matrix factorization for sequential 3d human poses estimation is proposed in [66]. In terms of occlusions, [63], [64], [65] and [66] focus on handling self-occlusions in single-person pose estimation, while our work addresses multi-person pose estimation with inter-person occlusions.

The advantage of two-stage methods include that (a) 2D pose are easy to predict with large amount of available labelled 2D pose estimation data, since 2D ground truth data is much cheaper to obtain compared to 3D ground truth; (b) 2d-to-3d methods are not sensitive to diverse scenarios and environments. Along the same line of research, we estimate 3D pose from 2D intermediate results in a hierarchical fashion, where we apply a prior human model that allows proportions of bone lengths to be adaptively determined.

The challenges of two-stage methods, on the other hand, is that the two-stage solutions are subject to the 2D pose estimation performance. For challenging real-world scenarios with heavy occlusions in the videos, especially for natural multi-person interaction scenes, 2D detectors could give unreliable 2D keypoints.

While 3D human pose estimation of a single person based on monocular moving camera has been reported in these work, there are less papers that have analyzed 3D multi-human pose estimation performance in natural videos, especially recordings from car-mounted cameras, which are of high interest to autonomous driving, etc. A few papers show results for in-the-wild data such as MPII and MPII-INF-HP [30] but do not have 3D quantitative results on multi-person videos. Our work, on the other hand, addresses 3D multi-person human pose estimation in a variety of natural videos including transportation scenario, daily activities, sports, etc, which shows good results in both qualitative and quantitative results. We will compare our method with several state-of-the-art 2D-to-3D methods. Our aim is not to show increased accuracy on well-trained dataset recorded in experimental setting, which can be better handled by deep learning approaches, as we believe there are certain limitations in this branch of work. Instead, we show superior performance on natural multi-person videos recorded by moving cameras.

2.3. *Datasets for 3D Human Pose Estimation*

2.3.1 *Related Work*

The most commonly used datasets for 3D human pose estimation include Human3.6M [5], HumanEva [56], TotalCapture [57], MPI-INF-3DHP [58] and 3DPW [59]. Human3.6M is a large-scale dataset featuring single person at the center of videos recorded by commercialized MoCap system. MoCap system is considered the most accurate system for obtaining 3D ground truth in the literature. However, the cameras in the MoCap system are static. HumanEva is similarly captured by a MoCap system much earlier and contains simpler poses. TotalCapture captures diverse body motion using a massively multi-view sensor system, where the sensors are also statically installed. MPI-INF-3DHP tries to include more challenging poses in the

wild, but still focuses on single person and static cameras. 3DPW is the first dataset that includes video footage taken from a moving phone camera. However, it uses IMUs to record ground truth, which is reported to possess a 26mm average 3D pose error when verified with the Mocap system.

Unlike existing 3D human pose estimation datasets, our new dataset, Moving camera Multi-Human interactions (MMHuman) dataset, is aimed at providing accurate MoCap ground truth for multi-person interaction videos captured by moving cameras, and intentionally includes occlusions and human interactions that are common in real-world scenarios. We do not see our proposed dataset as an alternative to existing datasets; rather, MMHuman complements existing ones with accurate ground truth of new multi-person interaction sequences shot by freely moving cameras. Our new dataset allows a quantitative evaluation of state-of-the-art approaches for multi-human interaction cases that have heavy and inter-person occlusions.

2.3.2 *Reasons We Need New Dataset*

a) For evaluation.

The state-of-the-art methods try to estimate 3D human pose for multiple people in the wild. However, quantitative evaluation is still lacking due to lack of systematic multi-person moving camera dataset. 3DPW include some data in the wild using moving camera. However, the ground truth it provides was recorded by IMU combined with camera, reporting a 26mm average 3D pose error. Besides that, 3DPW does not focus on human-human interactions. Therefore, a new dataset of natural multi-person interactions with accurate ground truth is still in need.

b) For future research

For the benefit of future research, new dataset is needed for training. Training based methods for 3D human pose understanding require training data that is only available through Motion Capture (MoCap) systems [42], [5], [43]. Even if they show accurate pose estimation results (including occluded joints) in controlled environments, these approaches do not generalize well to real images, with the exception of recent work based on data synthesis that shows promising results in the wild [44], [45]. Synthetic data have proved to be useful for training CNN architectures, yet often requiring a domain adaptation stage [40] However, none is realistic enough in terms of clothing, hair or interactions with objects to be considered as a fully-convincing alternative to real images. Therefore, new dataset will be helpful. Current datasets are lacking in muti-human interactions and moving camera variations.

3. A New Moving Camera Multi-Human Interaction Dataset

3.1. *Features of the New Dataset*

The new Moving camera Multi-Human interaction (MMHuman) dataset has several unique features compared to existing datasets.

First, to our best knowledge this is the first multi-person moving camera dataset that provides accurate MoCap ground truth.

Second, MMHuman include common human interaction scenes where both inter-person and intra-person occlusions are common.

3.2. *New Dataset Design*

3.2.1 *Environment Setting*

The new dataset we collected consist of commercialized motion capture (MoCap) system and additional moving cameras. The MoCap system consist of several IR cameras. See Figure 3.1.

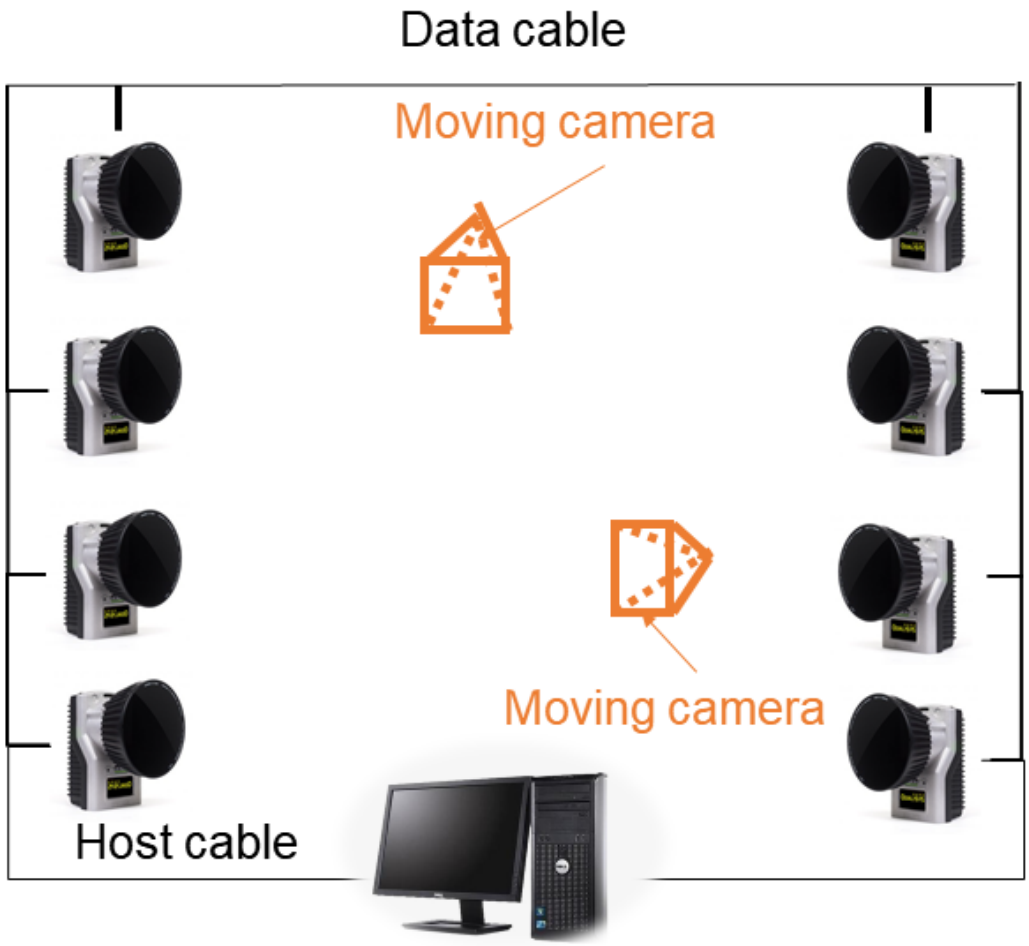


Fig. 3.1. Environment setting for dataset recording.

3.2.2 *Pose Classes*

We record some pose data that is lacking in the current literature, but is common in daily life. We emphasize on multiple people (human-human interaction), moving camera. In our data, there is serious amount of occlusion, which is the first dataset of such kind to our knowledge. Some examples of classes are shown in Fig. 3.4.

Our pose classed include:

(1) Shake hands

People shake hands with each other while talking to each other.

(2) Walking and crossing

People walking and crossing each other.

(3) High five

People high-five with each other.

(4) Hand Over

Hand over an object to one another.

(5) Kungfu

(6) Pull Up

One subject pull another sitting subject up.

(7) Basketball

Two or three actors playing basketball.

3.2.3 *Raw Data*

The frame rate of the moving camera videos is 30fps. The frame rate of MoCap capture is 120fps, but we resample to 30fps. Length of each video is roughly 1 minute. Therefore, each video has about 1800 frames.

A cleaned up subset of data, MMHuman20K, contains more than 20K frames of images. MMHuman20K provides 3D ground truth, 2D ground truth and pseudo 2D ground truth generated by OpenPose.

Another cleaned up subset of data, MMHuman50K, contains 50K frames of images. MMHuman50K provides 3D ground truth and pseudo 2D ground truth generated by OpenPose, but does not contain 2D ground truth.

3.2.4 *Obtaining 3D Ground Truth*

Step 1: Decide marker labels and attach them to subjects

We use 24 marker per person. The marker labels are listed below (shown in Figure 4.2)

R_AACR, L_AACR = right/left anterior acromion

R_PACR, L_PACR = right/left posterior acromion

R_HLE, L_HLE = right/left humeral lateral epicondyle

R_HME, L_HME = right/left humeral medial epicondyle

R_RSP, L_RSP = right/left radial styloid process

R_USP, L_USP = right/left ulnar styloid process

R_HIP, L_HIP = right/left hip

R_FLE, L_FLE = right/left femoral lateral epicondyle

R_FME, L_FME = right/left femoral medial epicondyle

R_ANKL, L_ANKL = right/left lateral ankle (lateral malleolus)

R_ANKM, L_ANKM = right/left medial ankle (medial malleolus)

FHD = forehead

BHD = back head

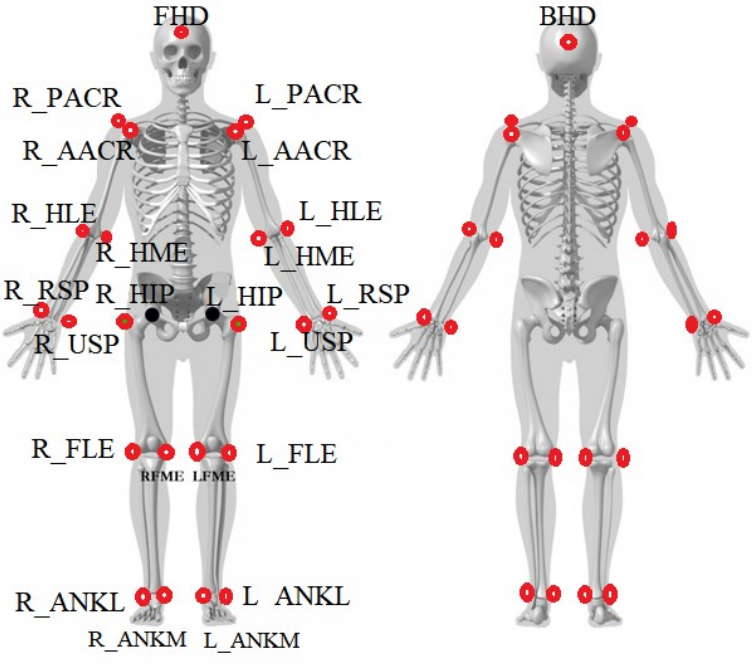


Fig. 3.2. Marker set.

Step 2: Record and save data

Record in MoCap system:

(1) Calibrate the system.

(2) Make sure every IR camera is functioning.

(3) Record, make sure people stay in the calibrated volume. Have an obvious (set of) action(s) for synchronization purpose.

(4) Save data.

Record using mobile camera:

Start recording at the same time, but with camera movement.

Step 3: Data cleaning

(1) Label each joint in the Qualysis software[48], according to label.

(2) When software loses track, re-label lost joints.

(3) Gap filling (linear or polynomial).

(4) Export data.

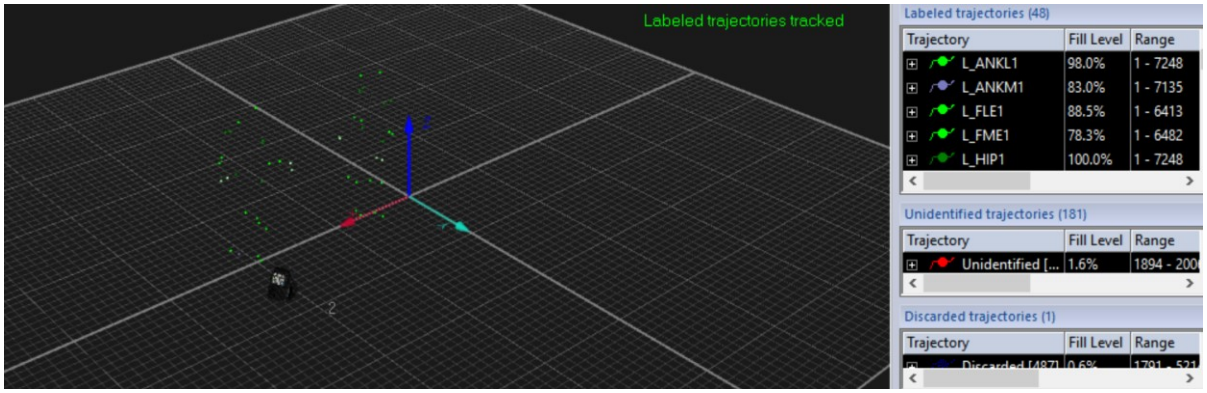


Fig. 3.3. Data cleaning.

a) Obtaining 2D Ground Truth

To provide a dataset for researchers, we will provide the accurate 2D ground truth. We will provide the labels of the 2D markers every 30 frames, in accordance with video frames. We will also provide pseudo ground truth, obtained by running OpenPose on the videos.

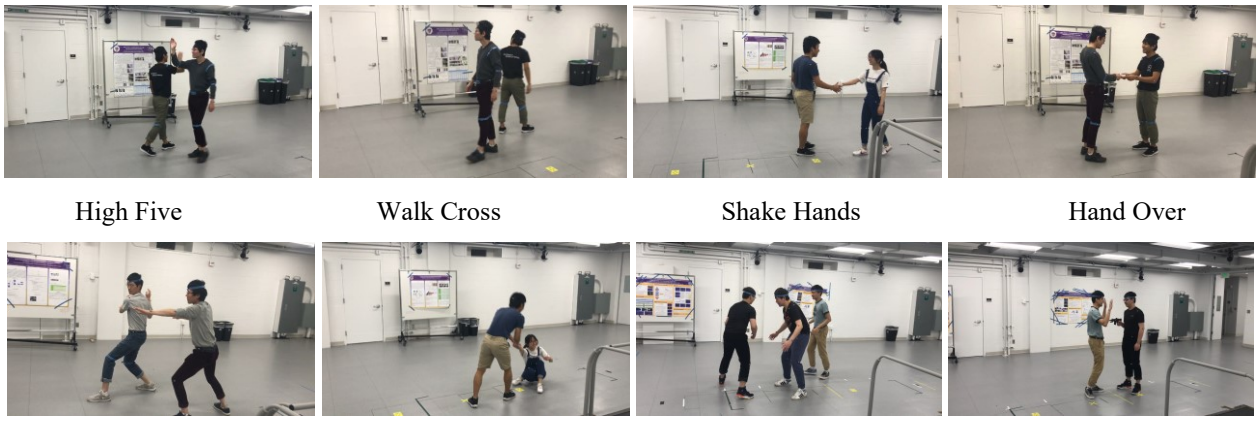


Fig. 3.4. MMHuman examples.

4. HIERARCHICAL 3D HUMAN POSE ESTIMATION

4.1. Overview of Hierarchical 3D Human Pose Estimation (H3DHPE)

The flowchart of our proposed Hierarchical 3D Human Pose Estimation (H3DHPE) is shown in Fig. 4.1. We first associate and track multiple people to use temporal information, and then estimating each person's 3D human poses hierarchically with body geometric constraints. *3D Localization,* including Visual Odometry (VO) and ground plane estimation, can be combined into the system to provide camera trajectory in the world coordinate and the human's real scale. When estimating each person's poses, we use 2D keypoints from OpenPose [1] as an intermediate step, and estimate 3D poses from the 2D keypoints. We apply a prior flexible human model that contains bone lengths of all human body parts, which can be optimized. Instead of trying to solve all the poses in high dimensions simultaneously, we first estimate the torso pose, and then estimate limb poses in a hierarchical fashion.

The contributions of the proposed Hierarchical 3D Human Pose Estimation (H3DHPE) is as follows.

(1) We propose a pipeline that integrates visual odometry, 3D human pose estimation, and exploit the temporal information using a powerful tracking method.

(2) We formulate the torso estimation as a Perspective-N-Point (PNP) problem and provide a highly efficient solution. For each limb, we formulate and solve an optimization problem. With the hierarchical problem solving structure, we greatly reduce complexity in high dimensional pose space. Moreover, each limb will not interfere with one another.

(3) We design an effective occlusion handling strategy utilizing keypoint confidence in case of missing or erroneous 2D human pose estimation.

(4) We provide a variety of experiments on natural videos containing multiple people recorded by static or moving monocular camera, which are lacking in the current literature but critical in today's applications. Our solution provides great opportunities to understand and predict human behaviors in natural videos.

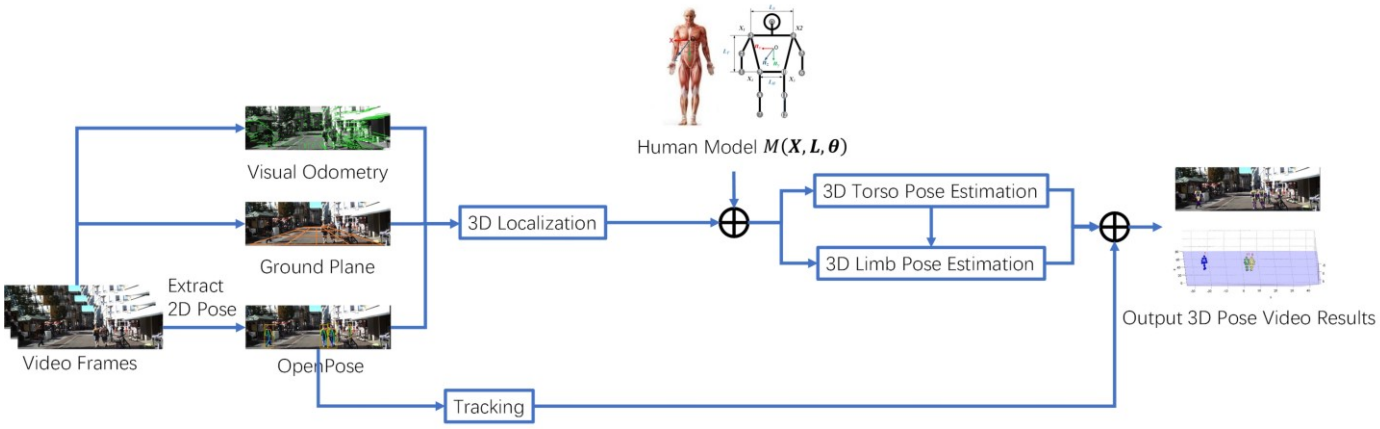


Fig. 4.1. Overall flowchart of the proposed system.

4.2. 2D Pose Tracking and 3D Localization by Moving Camera

4.2.1 *Notations*

First, as summarized in Table 4.1, we define some notations of our pose estimation system, where $(\cdot)_t$ represents the corresponding variable at the time t.

TABLE 4.1
NOTATIONS OF THE SYSTEM

Symbol	Notations
$\boldsymbol{R}^{(C)}$	Rotation matrix of the camera pose, 3×3
$\boldsymbol{t}^{(C)}$	Translation vector of the camera pose, 3×1
$\boldsymbol{R}^{(H)}$	Rotation matrix of the human pose, 3×3
$\boldsymbol{t}^{(H)}$	Translation vector of the human pose, 3×1
$\boldsymbol{K}$	Intrinsic camera matrix, 3×3
$\boldsymbol{n}^{(G)}$	Normal vector of the ground plane, 3×1
$h^{(G)}$	The camera height to the ground plane, scalar
$\boldsymbol{P}$	3D point in the world coordinate, (X, Y, Z)
$\boldsymbol{p}$	2D point on the image plane, $(x, y, 1)$
$\boldsymbol{X}^{(H)}$	3D joint in the human model coordinate, (X, Y, Z)
$\boldsymbol{X}^{(C)}$	3D joint in the camera coordinate, (X, Y, Z)
$\boldsymbol{X}^{(W)}$	3D joint in the world coordinate, (X, Y, Z)
$\boldsymbol{x}$	2D joint on the image plane, $(x, y, 1)$

4.2.2 *2D Pose Estimation and Tracking*

We take advantage of the recent advances in 2D human pose estimation using a deep neural network (DNN) based human pose detector, OpenPose [1], which independently detects 2D joints for every image frame of

the video. Since the 2D pose estimation is not the main focus of this paper, we just take the 2D pose estimated by OpenPose as the input initial value of our system.

OpenPose only focuses on single images. However, there are several drawbacks of single frame-based pose estimation, which are listed as follows.

a) The occlusion cannot be easily handled. When some joints are occluded, the single frame-based pose estimation becomes unreliable. If multiple people are close to each other, joints from different people can be easily tangled together. The pose cannot be estimated when full occlusion happens.

b) Temporal information is not well exploited for further analysis. Since there is no association across frames, it becomes unclear which person/joint corresponds to which person/joint across frames. Without association in the time domain, it is hard to perform further analysis, such as behavior analysis, anomaly detection and speed estimation.

To address the above drawbacks, a multiple-object tracking method is applied to solve occlusion and association problem. We adopt TrackletNet Tracking (TNT) [31] for human tracking under fixed or moving camera since it shows great performance when handling occlusions. Based on the detection results from OpenPose, the tracklets are generated based on intersection-over-union (IOU) and appearance similarity between two adjacent frames. Then the tracklet based graph model is built and we treat with each tracklet being treated as a node in the graph. For every edge between two nodes, the connectivity is defined measured by a the pre-trained multi-scale TrackletNet, which measures the similarity between two tracklets. Then clustering is conducted to minimize the total cost on the graph,. After clustering, so that the tracklets from the same ID can be merged into one group. The details of TNT can be found in [31]. Simply put, the goal of tracking is to obtain the unique ID for each detected person from OpenPose, i.e.,

$$id(D_{i,t}) = \mathcal{T}(D_{i,t}), \tag{4.1}$$

where $D_{i,t}$ is the i-th detection at Frame t and $\mathcal{T}(\cdot)$ is the tracker function. The output of the tracking is the unique person ID of the detection $D_{i,t}$. Since the detection results from OpenPose sometimes are very noisy, Kalman filter is applied for smoothing for each individual joint.

4.2.3 *3D Localization by Visual Odometry and Ground Plane Estimation*

We use state-of-the-art semi visual odometry (SVO) [19] technique to calculate the camera trajectory, so as to infer the 3D location of the human to be estimated. From SVO we can localize the camera position and pose, as well as the ground plane. Subsequently, the foot location for each human in the world coordinates can be estimated. For every 3D human-foot point $\boldsymbol{P}$ of the OpenPose detected person, who stands on the ground plane, the following two constraints should be followed,

$$\boldsymbol{K}\left(\boldsymbol{R}^{(C)}\boldsymbol{P}+\boldsymbol{t}^{(C)}\right)=s\boldsymbol{p}, \tag{4.2}$$

$$\boldsymbol{n}^{(G)}\boldsymbol{P}+h^{(G)}=0, \tag{4.3}$$

where the camera pose $\left[\boldsymbol{R}^{(C)}|\boldsymbol{t}^{(C)}\right]$ can be estimated by SVO [32], while the ground plane $\left(\boldsymbol{n}^{(G)},h^{(G)}\right)$ can be estimated by [33]. Note that, $\boldsymbol{p}$ is the 2D projection of the 3D point $\boldsymbol{P}$ on the image plane with scale factor s, as indicated in Eq. (3.2); Eq. (4.3) specifies the ground plane constraint. The 3D point $\boldsymbol{P}$ can thus be localized as a function of $\boldsymbol{R}^{(C)}$, $\boldsymbol{t}^{(C)}$, $\boldsymbol{n}^{(G)}$, and $h^{(G)}$(as defined in Table 4.1), derived from Eq. (4.2) and Eq. (4.3), i.e.,

$$\boldsymbol{P}\left(\boldsymbol{R}^{(C)},\boldsymbol{t}^{(C)},\boldsymbol{n}^{(G)},h^{(G)}\right)=\left(\boldsymbol{K}\boldsymbol{R}^{(C)}\right)^{-1}\left(\frac{\boldsymbol{n}^{(G)^T}\left(\boldsymbol{K}\boldsymbol{R}^{(C)}\right)^{-1}\boldsymbol{K}\boldsymbol{t}^{(C)}-h^{(G)}}{\boldsymbol{n}^{(G)^T}(\boldsymbol{K}\boldsymbol{R}^{(C)})^{-1}\boldsymbol{p}}\boldsymbol{p}-\boldsymbol{K}\boldsymbol{t}^{(C)}\right), \tag{4.4}$$

Then the absolute height, H, of the estimated human can be obtained by,

$$H=\frac{\|\boldsymbol{P}_{bl}-\boldsymbol{P}_{br}\|h}{w}, \tag{4.5}$$

where $\boldsymbol{P}_{bl}$ and $\boldsymbol{P}_{br}$ are the two bottom 3D points corresponding to the two bottom points of the detection bounding box on the ground plane, and w and h are the width and height of the 2D detection bounding box.

4.3. Proposed 3D Human Pose Estimation Method

4.3.1 *Flexible Hierarchical 3D Human Body Model*

To estimate reasonable 3D poses for human, we adopt a flexible hierarchical 3D human body model. The torso is at the top level in the human body model hierarchy. The upper limbs are at the second level, which depends on the top-level pose of the torso. The lower limbs are at the third level, which depends on the second-level upper limbs, and thus on the top-level pose of the torso. The human body model is defined as $M(\boldsymbol{X},\boldsymbol{L},\boldsymbol{\theta})$, parameterized by joints $\boldsymbol{X}$, bone lengths $\boldsymbol{L}$, and angles $\boldsymbol{\theta}$. In our definition of the human model, all the joints are flexible, which can be decomposed to different poses. However, the joints are also constrained by the bone length $\boldsymbol{L}$ and joint angle $\boldsymbol{\theta}$. In particular, there are 13 joints used in our human models. In the human model coordinate system, the origin is defined as the center of the torso plane, which also determines the 3D locations of shoulder and hip joints. An example of the flexible human model is shown in Fig. 4.2.

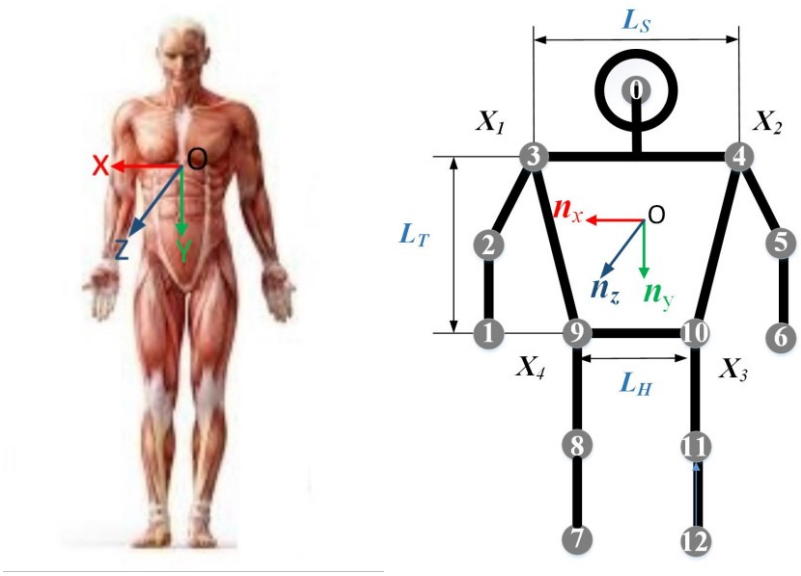


Fig. 4.2. Flexible hierarchical 3D human body model coordinates.

TABLE 4.2

ANGLE CONSTRAINTS OF THE DEFINED HUMAN MODEL

joint	axis	θ^-(degree)	θ^+(degree)
shoulder	$\boldsymbol{n}_x$	-60	180
	$\boldsymbol{n}_y$	-30	135
	$\boldsymbol{n}_z$	0	180
elbow	$\boldsymbol{n}_x$	-10	150
hip	$\boldsymbol{n}_x$	-15	140
	$\boldsymbol{n}_z$	-45	30
knee	$\boldsymbol{n}_x$	-10	150

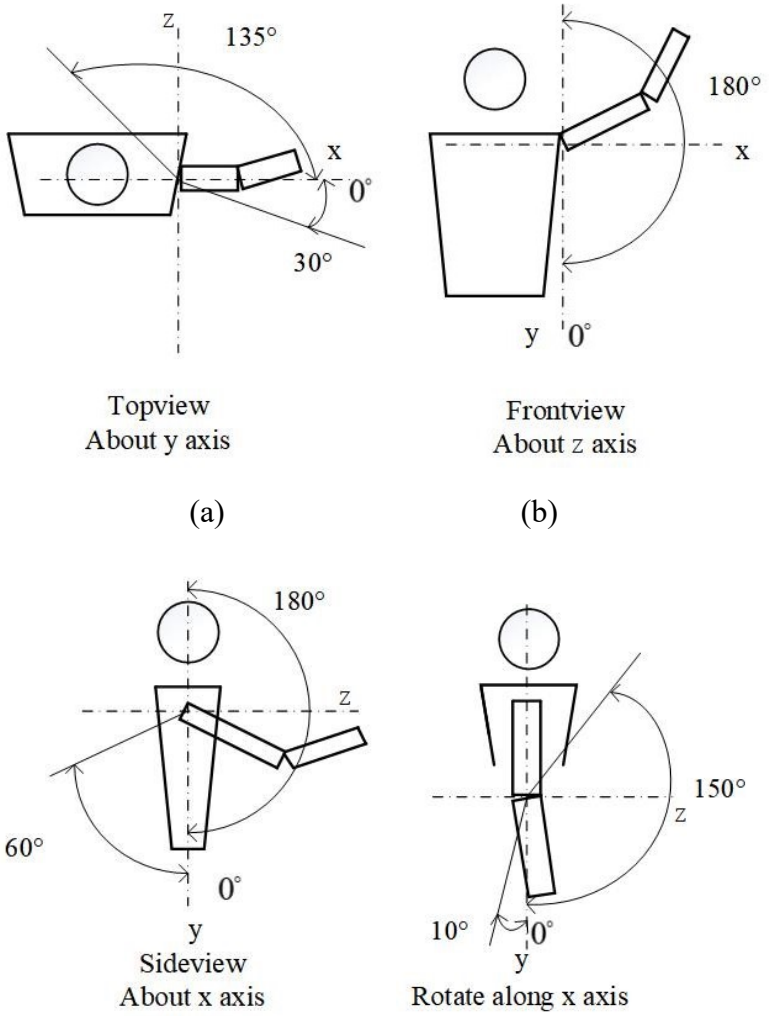
Then a connectivity matrix is defined as follows to measure whether two joints $\boldsymbol{X}_i$, $\boldsymbol{X}_j$ are connected in the skeleton,

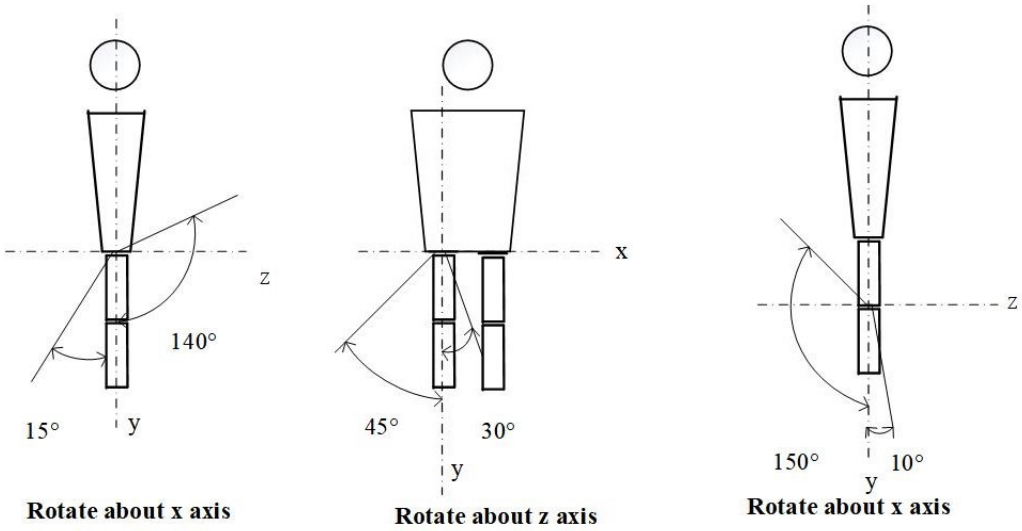
$$\boldsymbol{C}(i,j) = \begin{cases} 1 & \boldsymbol{X}_i, \boldsymbol{X}_j \; connected \\ 0 & \text{O.W.} \end{cases} \tag{4.6}$$

Similarly, the matrix of bone length between each pair of joints is defined as,

$$\boldsymbol{L}(i,j) = Hl_{i,j}, \tag{4.7}$$

where $l_{i,j}$ is the initialized normalized bone length between connected joints $\boldsymbol{X}_i, \boldsymbol{X}_j$. Note that bone lengths $\boldsymbol{L}(i,j)$ are proportional to the body height. We use this set of bone lengths to initialize, and then include bone lengths as parameters for optimization. This way, we can obtain a fine-tuned final estimation.





(e) (f) (g)

Fig. 4.3. Limb angle limits [38]. (a) Shoulder angle limits, top view. (b) Shoulder angle limits, front view. (c) Shoulder angle limits, side view. (d) Elbow angle limits. (e) Hip angle limits, side view. (f) Hip angle limits, front view. (g) Knee angle limits, side view.

Angle constraints are summarized in Table 4.2 and Fig. 4.3. Denote the 3D coordinate system of torso pose as $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, $\boldsymbol{n}_z$ as shown in Fig. 4.2. Denote $\boldsymbol{\theta}$ as the set of angle constraints. For any $u_{i,j} \in \boldsymbol{\theta}$, u is a 6-D tuple, denoted as $u_{i,j} = \left(i, j, \boldsymbol{n}_{i,j,1}, \boldsymbol{n}_{i,j,2}, \theta_{i,j}^{-}, \theta_{i,j}^{+}\right)$, where i, j are the pair of joints from one of the above seven categories, $\boldsymbol{n}_{i,j,1}$ and $\boldsymbol{n}_{i,j,2}$ are two axes of angle rotation plane picked from $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, $\boldsymbol{n}_z$, which are shown in each sub-plot of Fig. 4.3, and $\theta_{i,j}^{-}, \theta_{i,j}^{+}$ are the lower and upper bounds of the angle constraints. Denote $u_{i,j} = \left(x_{i,j,1}, x_{i,j,2}\right)$ as the 2D coordinate of the angle rotation plane, i.e.,

$$x_{i,j,1} = \frac{\boldsymbol{n}_{i,j,1}^{T}\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)}{\left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|}, \tag{4.8}$$

$$x_{i,j,2} = \frac{\boldsymbol{n}_{i,j,2}^{T}\left(\boldsymbol{X}_i - \boldsymbol{X}_j\right)}{\left\|\boldsymbol{X}_i - \boldsymbol{X}_j\right\|}. \tag{4.9}$$

Then the angle constraint can be represented as

$$\theta_{i,j}^{-} \leq \mathrm{Angle}\left(x_{i,j,1}, x_{i,j,2}\right) \leq \theta_{i,j}^{+}, \tag{4.10}$$

where $\mathrm{Angle}(x, y)$ is the function of x, y that outputs the angle of the point (x, y) on the 2D space.

4.3.2 *Hierarchical 3D Human Pose Estimation*

Our proposed pose estimation is processed in a hierarchical way, i.e., the torso pose is estimated first, then comes the upper and lower limb pose estimation. Several advantages of this hierarchical approach are discussed below.

- **Robustness in multi-person localization and pose estimation**. As we know, the four torso joints are least flexible than the joints on the limbs, which can be treated as a rigid object, and they usually form a 3D regular plane. Based on the Perspective-N-Point (PNP) algorithm [34], the torso poses can be estimated and localized related to the camera easily and robustly. On the other hand, if we also take into account the joints on the limbs at this point, then the 3D torso pose cannot be easily and accurately inferred.
- **Simplify the estimation**. Since the limb pose is largely dependent on the torso pose, after we estimate the torso pose, the limb pose can be easily inferred. Moreover, the dissection of the problem greatly reduces the search space with increased efficiency. Methods that try to solve the full set of poses, i.e., 13 joints, suffer from computation complexity. On the contrary, we structure poses hierarchically and formulate torso and limb estimations respectively with a lower degree of freedoms (DoFs), enabling real-time processing capability. This hierarchical way can largely simplify the constraints and make the optimization efficiently.

a) Torso Pose Estimation

For each person, the camera pose can be inferred by solving a PNP problem with four pairs of 3D torso joints in the human model and 2D joints on the image plane. For each 3D and 2D pair i of the k-th person, they should follow the projection constraint as follows,

$$s\boldsymbol{x}_{k,i} = \boldsymbol{K}\left(\boldsymbol{R}_k^{(H)}\boldsymbol{X}_{k,i}^{(H)} + \boldsymbol{t}_k^{(H)}\right), \tag{4.11}$$

where $\boldsymbol{X}_{k,i}^{(H)}$ represents the location of the i-th torso joint of the k-th person in the human model coordinate. Particularly, we use $\boldsymbol{X}_{k,i\in\{1,2,3,4\}}$ to denote the torso points, as shown in Fig. 4.2. For simplification, we denote L_S, L_T and L_H as the length of shoulder, torso, and hips, respectively as shown in Fig. 4.2. Then, the four torso points can be represented as

$$\boldsymbol{X}_{k,1} = \left(\frac{L_S}{2}, -\frac{L_T}{2,}, 0\right), \boldsymbol{X}_{k,2} = \left(-\frac{L_S}{2}, -\frac{L_T}{2}, 0\right),$$

$$\boldsymbol{X}_{k,3} = \left(-\frac{L_H}{2}, \frac{L_T}{2}, 0\right), \boldsymbol{X}_{k,4} = \left(\frac{L_H}{2}, \frac{L_T}{2}, 0\right). \tag{4.12}$$

Given 4 pairs of joints, the torso pose $\left[\boldsymbol{R}_k^{(H)} | \boldsymbol{t}_k^{(H)}\right]$ of the k-th person in the camera coordinate can be represented by

$$\boldsymbol{X}_{k,i}^{(C)} = \boldsymbol{R}_k^{(H)} \boldsymbol{X}_{k,i}^{(H)} + \boldsymbol{t}_k^{(H)}. \tag{4.13}$$

We can also transform the world coordinates to the camera coordinates with the estimated camera pose by

$$\boldsymbol{X}_{k,i}^{(C)} = \boldsymbol{R}^{(C)} \boldsymbol{X}_{k,i}^{(W)} + \boldsymbol{t}^{(C)}, \tag{4.14}$$

Then, we can get the torso joint $\{\boldsymbol{X}_{k,i}^{(W)}\}$ in the world coordinate as

$$\begin{aligned} \boldsymbol{X}_{k,i}^{(W)} &= \left(\boldsymbol{R}^{(C)}\right)^{-1} \boldsymbol{X}_{k,i}^{(C)} - \left(\boldsymbol{R}^{(C)}\right)^{-1} \boldsymbol{t}^{(C)} \\ &= \left(\boldsymbol{R}^{(C)}\right)^{-1} \left(\boldsymbol{R}_k^{(H)} \boldsymbol{X}_{k,i}^{(H)} + \boldsymbol{t}_k^{(H)}\right) - \left(\boldsymbol{R}^{(C)}\right)^{-1} \boldsymbol{t}^{(C)}. \end{aligned} \tag{4.15}$$

b) Limb Pose Estimation

The limb pose estimation is built upon the estimated torso pose, which is one important feature of our hierarchical pose estimation. The limb pose is estimated based on the reprojection error as well as the constraints of bone length $\boldsymbol{L}(i,j)$, and joint angle $\boldsymbol{\theta}$, which are defined in the previous section. Then the cost function of the joints on any limb is defined as follows,

$$\begin{aligned} f\left(\boldsymbol{X}_{k,i}^{(C)}\right) = & \sum_i c_{k,i} \left\| \boldsymbol{K} \boldsymbol{X}_{k,i}^{(C)} - s_{k,i} \boldsymbol{x}_{k,i} \right\| \\ & + \rho_1 \sum_{u_{i,j} \in \boldsymbol{\theta}} d\left(\text{Angle}\left(x_{i,j,1}, x_{i,j,2}\right), R\left(\theta_{i,j}\right)\right) \\ & + \rho_2 \sum_{i,j} \boldsymbol{C}(i,j) d\left(\left\| \boldsymbol{X}_{k,i}^{(C)} - \boldsymbol{X}_{k,j}^{(C)} \right\|, R\left(\boldsymbol{L}(i,j)\right)\right), \end{aligned} \tag{4.16}$$

where $c_{k,i}$ is the confidence score of the i-th joint on the person k-th from OpenPose, $\boldsymbol{X}_{k,i}^{(C)}$ is one of the joints

of on the limb of the k-th person, and $u_{i,j}$ is a 6-D tuple element from $\boldsymbol{\theta}$ defined in the human model. For arms (upper limbs), it can be either elbow or wrist; for legs (lower limbs), it can be either knee or foot. $\boldsymbol{x}_{k,i}$ is the corresponding 2D joint on the image plane, $s_{k,i}$ is the scale, $R(\theta_{i,j}) = [\theta_{i,j}^{-}, \theta_{i,j}^{+}]$ is the range of the joint angle, $R(\boldsymbol{L}(i,j)) = [\boldsymbol{L}(i,j) - \delta_l,\ \boldsymbol{L}(i,j) + \delta_l]$ is the range of the bone length, where δ_l is a small amount. The distance function $d(x, R)$ measures the cost between x and R, i.e.,

$$d(x,R) = \exp\left(\min_{r \in R}\left(\frac{|x-r|}{\max(R) - \min(R)}\right)\right) - 1. \tag{4.17}$$

If x lies in the range R, then the output of the function is 0; otherwise, the output is the minimum exponential absolute distance to the range R.

$\rho_1$and ρ_2 are weights that normalize each term to be in the same range with the first term respectively.

The cost function in Eq. (4.16) can be efficiently solved by Powell's method [35]. After the optimization, the joint location in the world coordinate can be obtained by Eq. (4.15).

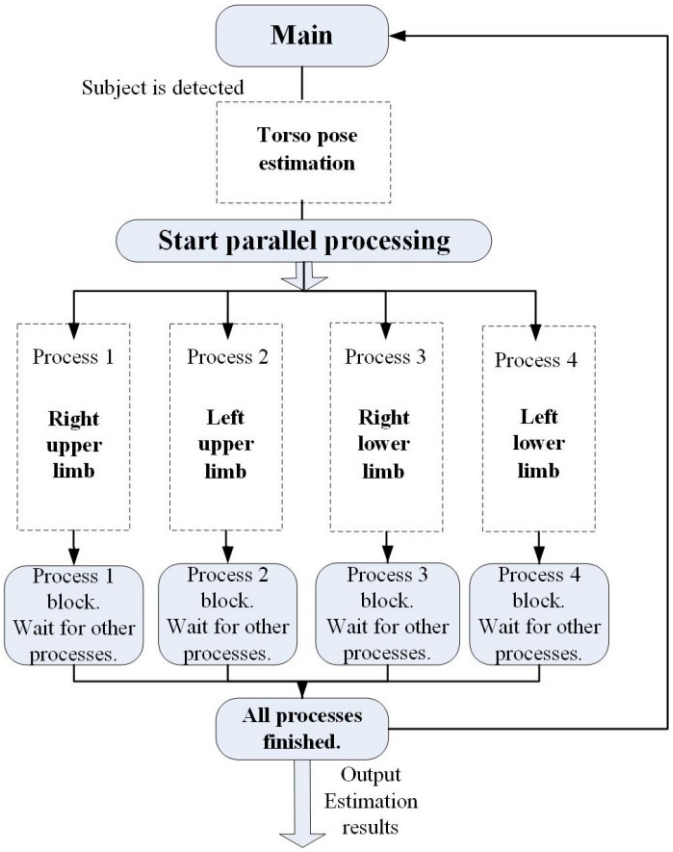


Fig. 4.4. Flowchart of parallel processing.

Thanks to our hierarchical human model, we are able to dissect the calculation and adopt parallel processing efficiently for the single frame pose initialization. Below is the flowchart of our limb parallel processing design for a single subject. For multiple persons, each person can be processed in parallel as well, which is shown in Fig. 4.4.

c) Occlusion Reasoning

We utilize keypoint confidence and geometry reasoning to handle occlusion. For the joint in the camera coordinate, if it is occluded, then the reprojection constraint needs to be relaxed since the point is hard to be seen from the camera view. In other words, we are interested in the probability that a joint is both detected and visible (not occluded), which can be formulated as

$$P(V, D) = P(V|D)P(D), \tag{4.18}$$

where $P(D)$ is the detection probability, i.e., the joint confidence $c_{k,i}$ computed by OpenPose, $P(V|D)$ is the probability of the visibility of the detected joint, and $P(V, D)$ is the probability that a joint is both detected and visible.

To approximate $P(V|D)$, a sigmoid function is adopted, i.e.,

$$P(V|D) = \frac{1}{1 + \exp(Z - Z_c)}. \tag{4.19}$$

Z is the depth of the target joint, and Z_c is the depth of the center of the torso. If Z is larger than Z_c, then the joint is more likely to be occluded by the torso; otherwise, it is more likely to be visible. This situation is very common especially when the camera is on the side view of the human body. Rather than using $c_{k,i}$ as the weight of reprojection error directly, we use $P(V, D)$ as the weight. Then Eq. (4.16) is reformulated as

$$\begin{aligned} f\left(\boldsymbol{X}_{k,i}^{(C)}\right) = & \sum_{i} P(V, D) \left\| \boldsymbol{K}\boldsymbol{X}_{k,i}^{(C)} - s_{k,i}\boldsymbol{x}_{k,i} \right\| \\ & + \rho_1 \sum_{u_{i,j} \in \boldsymbol{\theta}} d\left(\text{Angle}\left(x_{i,j,1}, x_{i,j,2}\right), R\left(\theta_{i,j}\right)\right) \\ & + \rho_2 \sum_{(i,j)} \boldsymbol{C}(i,j) d\left(\left\| \boldsymbol{X}_{k,i}^{(C)} - \boldsymbol{X}_{k,j}^{(C)} \right\|, R\left(\boldsymbol{L}(i,j)\right)\right). \end{aligned} \tag{4.20}$$

4.4. Joint Optimization with Temporal Constraints

After we initialize the torso pose and limb pose in a hierarchical way for each frame, we also want to take consideration of temporal constraints to handle missing poses and occlusions with smoothness constraints. Rather than estimating the joint location in the camera coordinate, we fine-tune the 3D joint location in the world coordinate directly since the joint locations are usually very smoothing in the world coordinate and independent to the camera pose. The cost function with temporal constraints involved is defined as follows,

$$
\begin{aligned}
f\left(\boldsymbol{X}_{i,t}^{(W)}\right) = & \sum_{t}^{T}\sum_{i}^{N} P(V,D)\left\|\boldsymbol{K}\left(\boldsymbol{R}_{t}^{(C)}\boldsymbol{X}_{i,t}^{(W)}+\boldsymbol{t}_{t}^{(C)}\right)-s_{i,t}\boldsymbol{x}_{i,t}\right\| \\
& +\lambda_1\sum_{t}^{T}\sum_{u_{i,j}\in\boldsymbol{\theta}} d\left(\text{Angle}\left(x_{i,j,1},x_{i,j,2}\right),R\left(\theta_{i,j}\right)\right) \\
& +\lambda_2\sum_{t}^{T}\sum_{(i,j)}\boldsymbol{C}(i,j)d\left(\left\|\boldsymbol{X}_{i}^{(W)}-\boldsymbol{X}_{j}^{(W)}\right\|,R\left(\boldsymbol{L}(i,j)\right)\right) \\
& +\lambda_3\sum_{t}^{T}\sum_{i}^{N}\left\|\boldsymbol{X}_{i,t}^{(W)}-\boldsymbol{X}_{i,t-1}^{(W)}\right\|^2, \qquad (4.21)
\end{aligned}
$$

Since we estimate the pose of each person individually, the person index k is dropped for simplification. In addition to the cost defined by Eq. (4.20), a temporal smoothness constraint is added as the last term of Eq. (4.21).

4.5. Experiments and Analysis

We test our method on several videos, such as Kitti dataset [2], ETH [3] and videos in DALY dataset [4] which show the qualitative evaluation. We also test videos in UWHHI [51] and Human3.6M [5] for quantitative evaluation. Moreover, ablation study is conducted on different parameter settings. At the time of this work, the new dataset MMHuman (See Chapter 3) has not been collected, therefore, H3DHPE results on MMHuman are not included.

4.5.1 *Experimental setting*

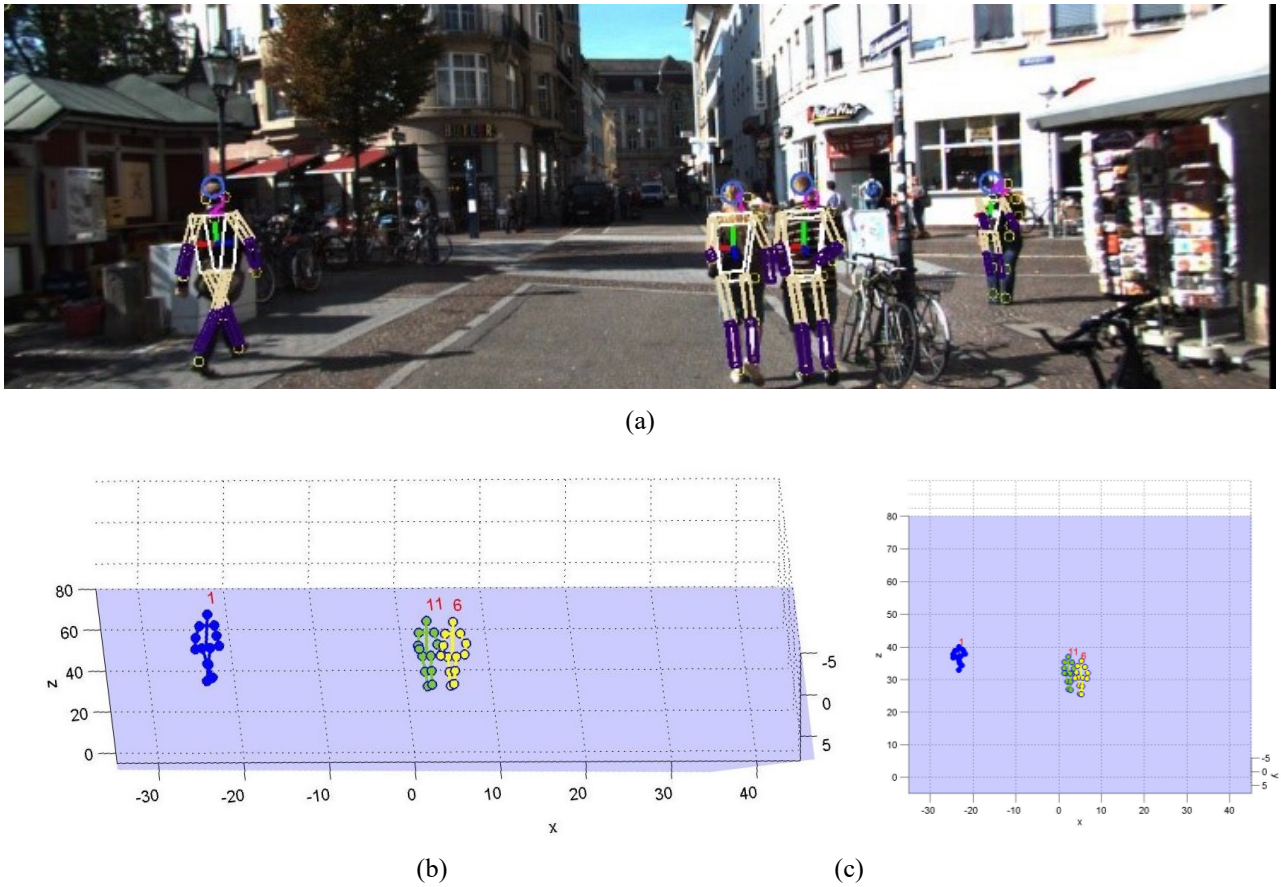
We tested our program on Windows 10, using an Intel Core i5-6300HQ CPU@2.30Ghz,2301Mhz, 4 Core Processor.

4.5.2 *Qualitative Evaluation*

a) KITTI Dataset

KITTI datasets are captured by driving around the mid-size city of Karlsruhe, in rural areas and on

highways. It is a highly popular dataset for autonomous driving research. We chose some sequences from this dataset to show our capability to estimate pedestrian poses using car-mounted monocular camera. Fig. 4.5 shows a snapshot of 3D pose estimation of pedestrians and cyclers based on our proposed scheme.

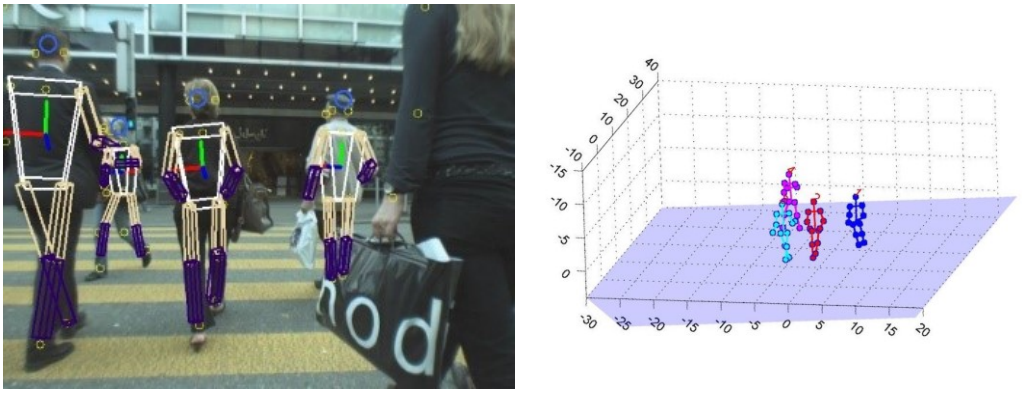


(a) (b) (c)

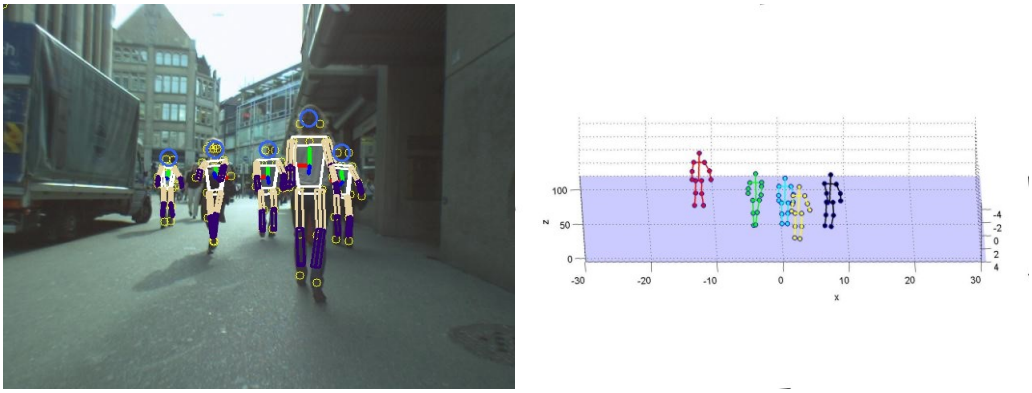
Fig. 4.5. 3D human pose estimation for multiple people on the street - kitti_2011_09_29. (a) Street video and human model reprojection. (b) Estimated 3D poses, front view. (c) Estimated 3D poses, top view.

b) ETH Dataset

ETH dataset is a dataset designed for challenging tasks of multi-person tracking. ETH dataset features transportation scenarios that contain dense pedestrians. Data was recorded using a pair of AVT Marlins F033C mounted on a chariot respectively a car, with a resolution of 640 x 480 (bayered), and a framerate of 13-14 FPS. We use only the left camera sequence. Our experiments show that we can also effectively perform multi-person 3D pose estimation.



(a)



(b)

Fig. 4.6. Results on ETH dataset. (a) 3D human pose estimation for multiple people on the street – ETH dataset CROSSING sequence. (b) 3D human pose estimation for multiple people on the street – ETH dataset LINTHESCHER sequence.

c) DALY dataset

DALY dataset is a dataset consists of daily activities. We find result of our method looks more natural and smoother compared to [24]. Furthermore, thanks to powerful 2D pose deep learning predictions, the method can handle occlusion to a certain extent. Fig. 4.7 shows screenshots of results for this video. Fig. 4.7 (b) and (c) show examples of successful 3D estimation with occluded hand and/or elbow.

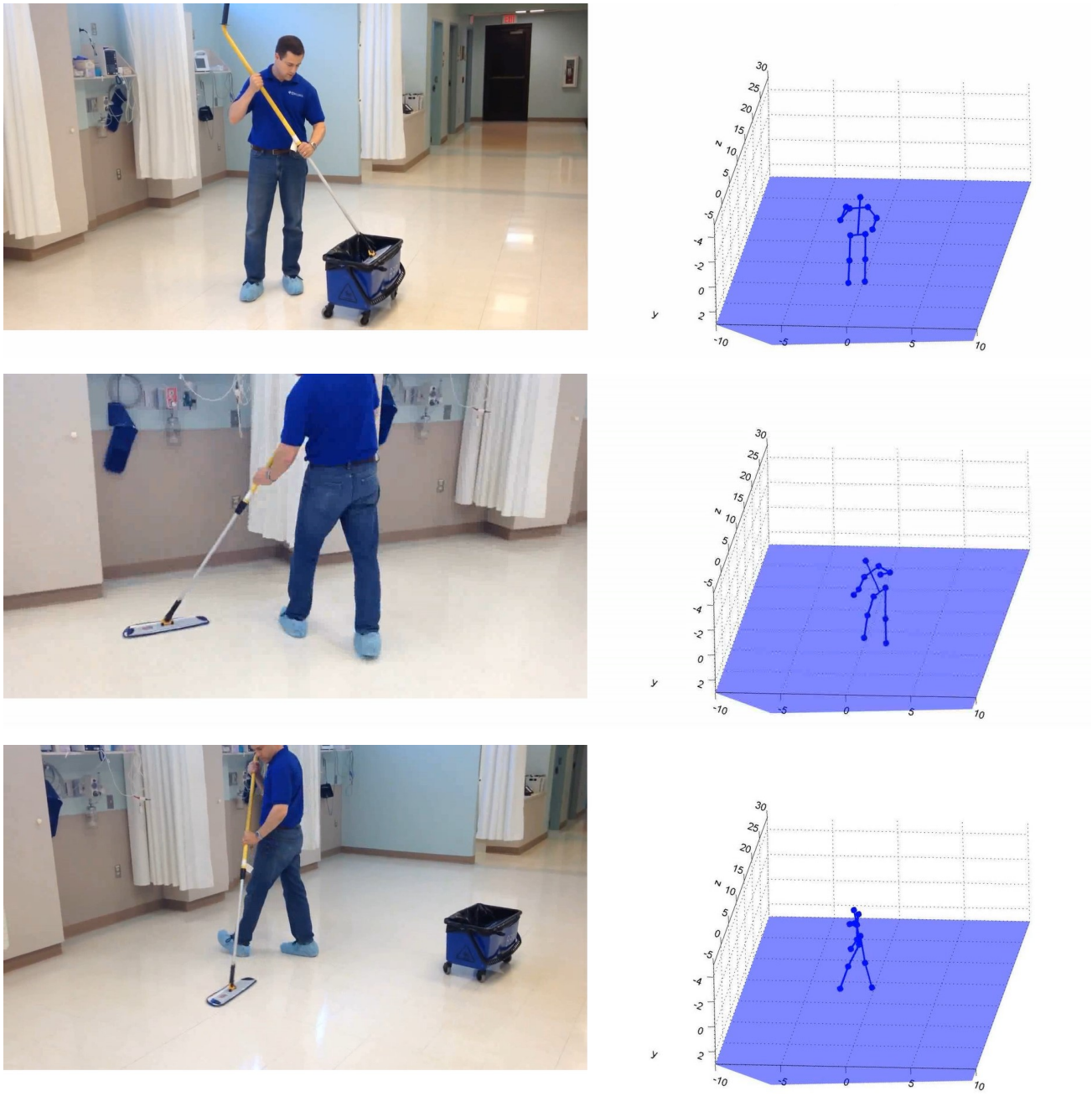
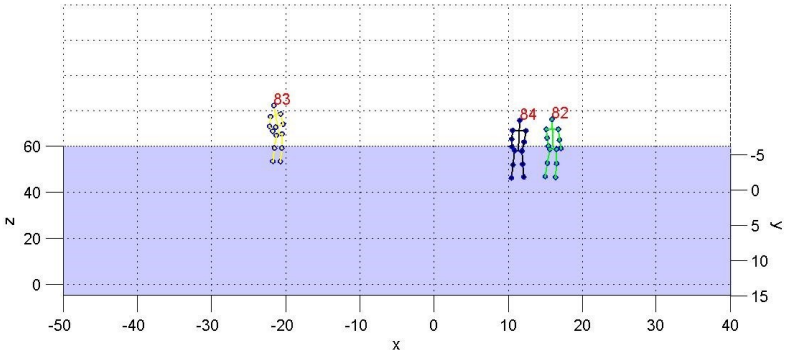


Fig. 4.7. Screenshots of DALY dataset “mop ground”.

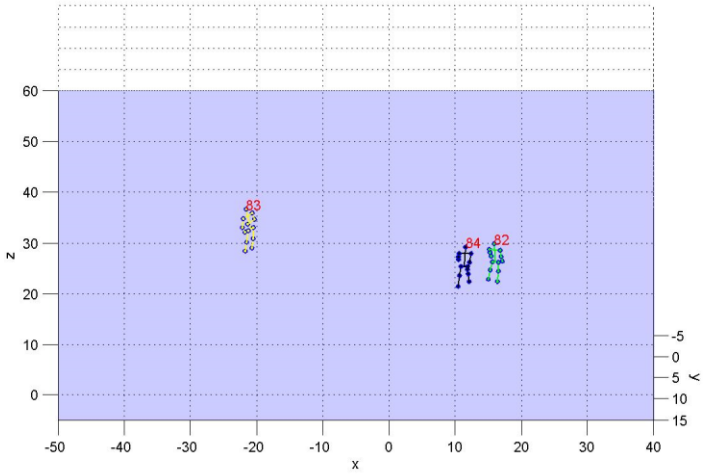
d) Public YouTube videos

We downloaded some public YouTube videos of street scenarios, and tested our method on these videos. Experiments show our method can handle these generic videos. Figure 4.8 shows screenshots of our results.

(a) Public YouTube video of street scenario



(b) Front view

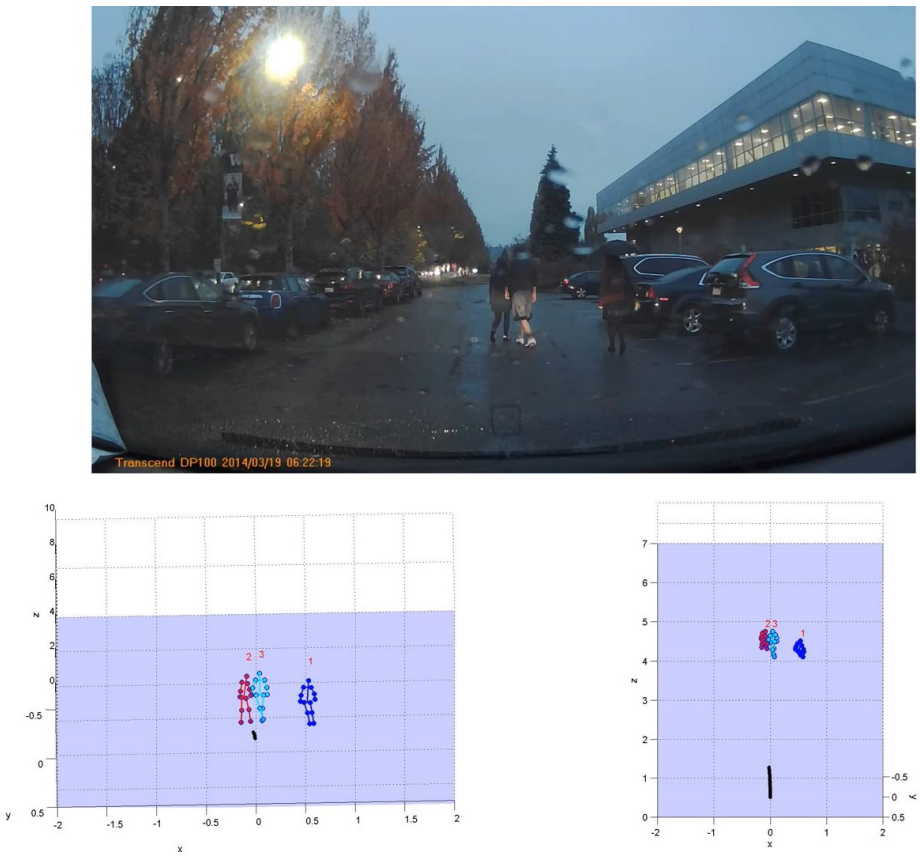


(c) Top view

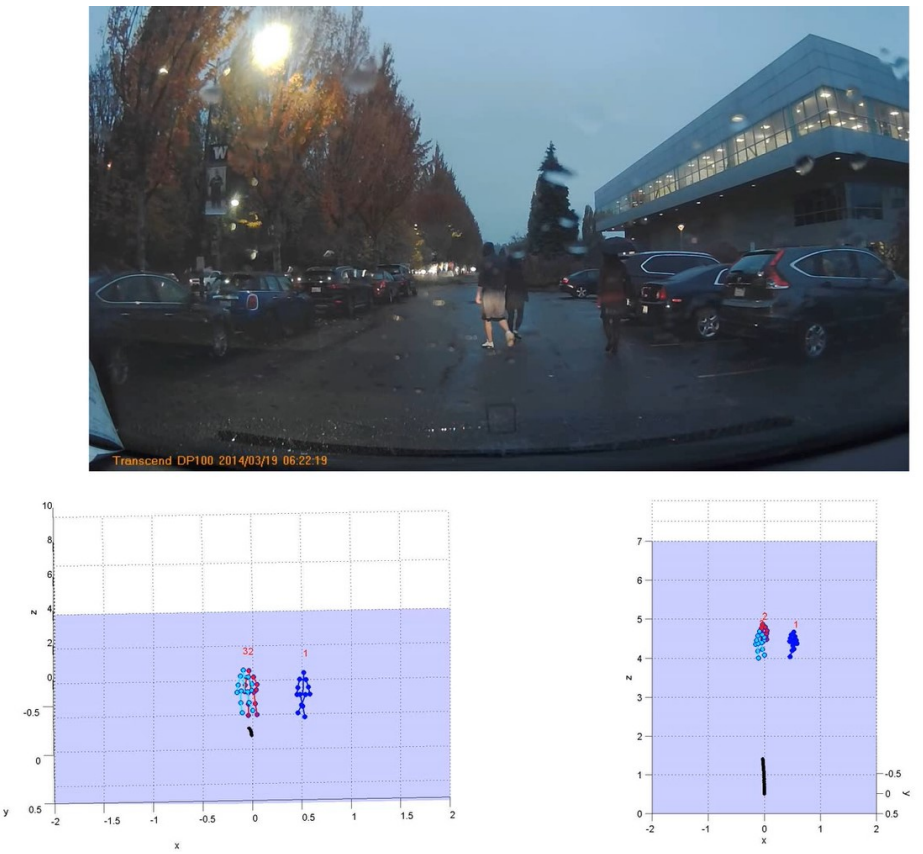
Fig. 4.8. Multi-person 3D human pose estimation for YouTube video of street scenario.

e) Parking Lot scenario

We recorded some real-world parking lot scenario video, and tested our method on these videos. Even with challenging weather and lighting condition, our method can track and estimate 3D human poses. Fig. 4.9 shows a case of occlusion when pedestrians are crossing each other. Our method can handle such cases pretty well.



(a) Parking Lot scenario: before crossing

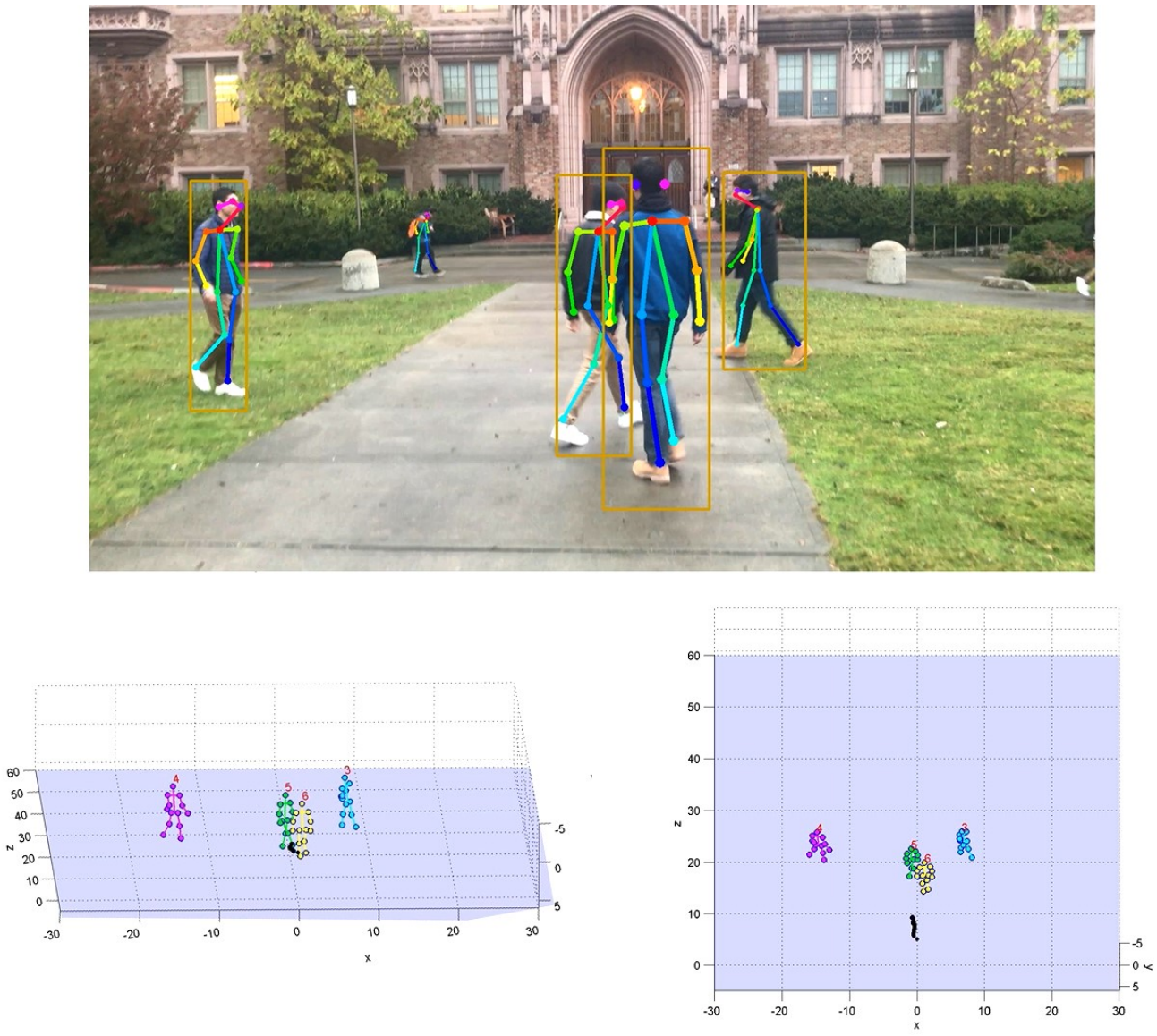


(a) Parking Lot scenario: after crossing

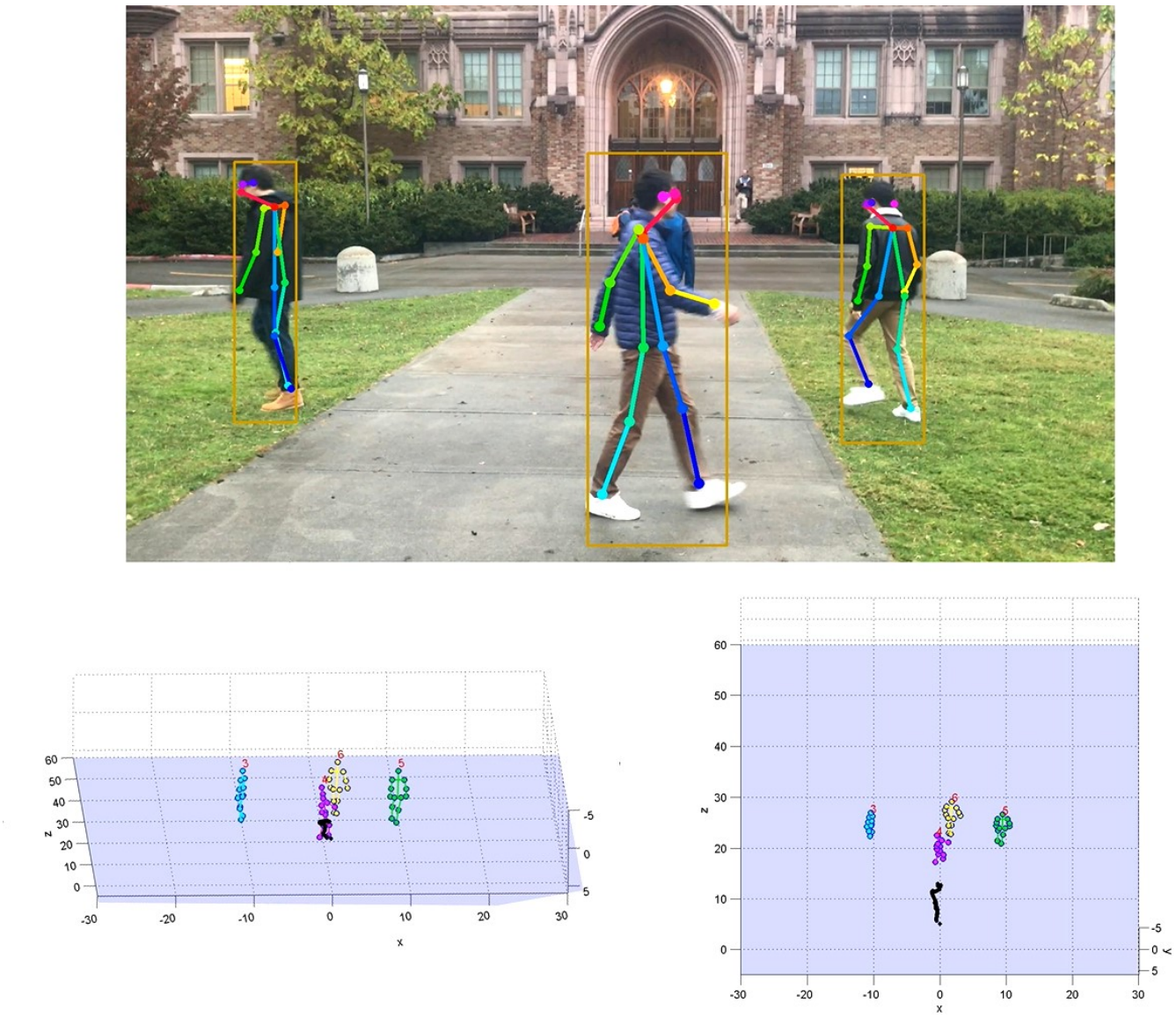
Fig. 4.9. Multi-person 3D human pose estimation for Parking Lot scenario.

f) Campus scenario

We recorded some pedestrian videos on campus, and tested our method on these videos. Fig. 4.10 shows a screenshot of 3D human pose estimation for crowd. Again, our method can handle cases with lots of occlusions.



(a) Result on campus crowd.
Top: OpenPose results. For accuracy purposes, pedestrians that are too far away are not displayed. Bottom Left: front view. Bottom right: top view.



(b) Campus crowd. Top: OpenPose results. Bottom Left: front view. Bottom right: top view.

Fig. 4.10. Multi-person 3D human pose estimation for Campus Walk scenario.

4.5.3 *Quantitative Evaluation*

a) UWHHI

We recorded multi-person moving camera data with ground truth, using Kinect One, as such dataset is lacking in the literature. Horizontal Resolution of Kinect One is 0.75 mm per pixel x by y at 0.5 m, and 3 mm per pixel x by y at 2 m. Depth resolution is about 1.5 mm at 0.5 m, and 3 mm at 3 m. We refer to it as University of Washington Human Human Interaction (UWHHI) data. To the best of our knowledge, none of the state-of-the art methods report quantitative evaluation on multi-person using monocular moving camera. Fig. 4.11 and Fig. 4.12 show comparisons to state-of-the-art deep learning method hg3d [30]. As shown in

Table 4.3, although hg3d shows higher performance for Human3.6M dataset, our method outperforms hg3d for multi-person human pose estimation using moving camera in natural scenarios.

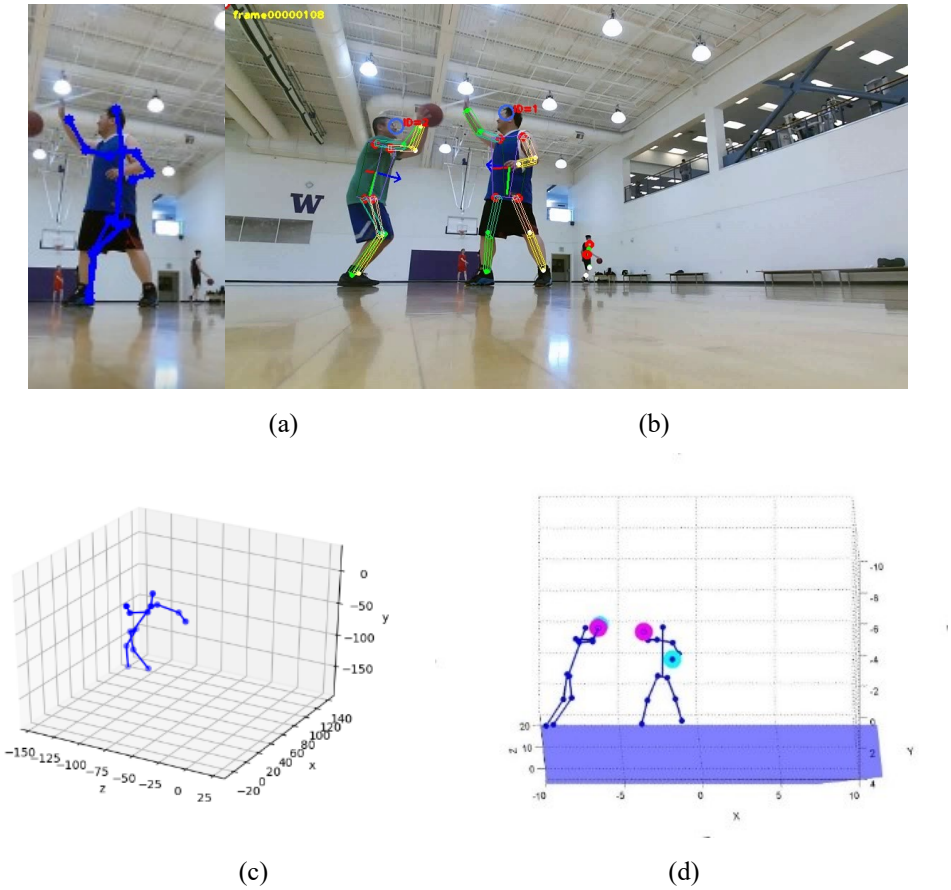
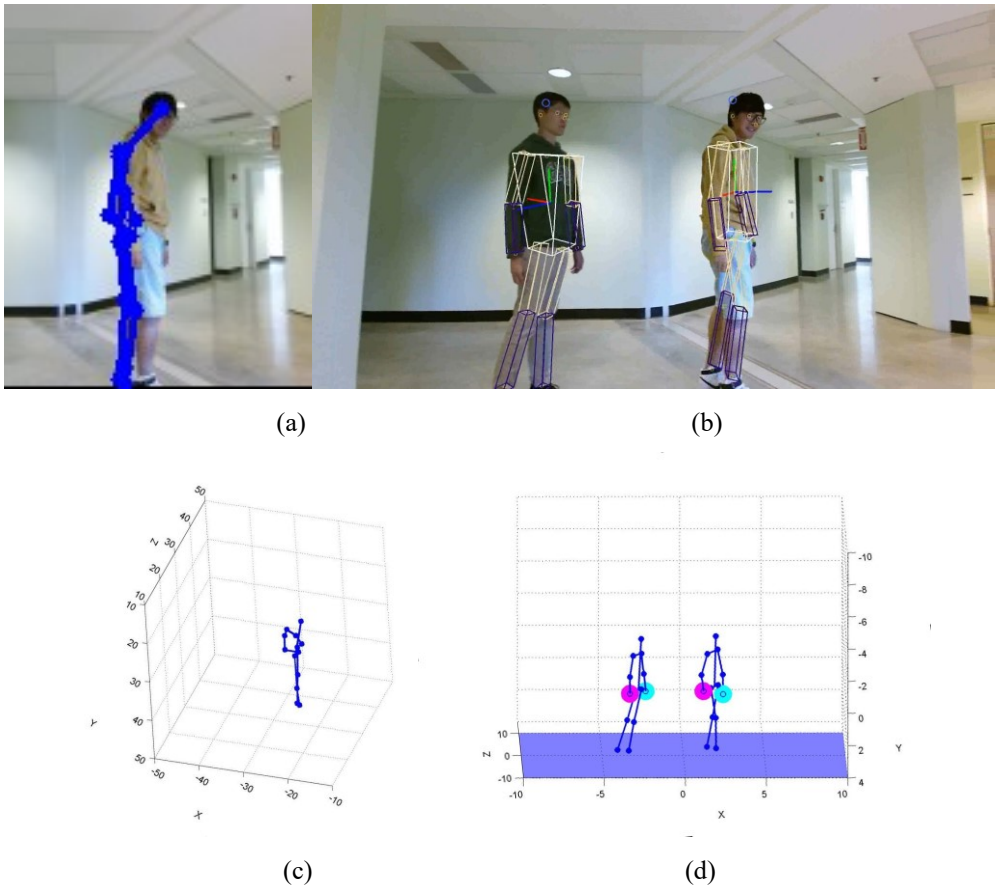


Fig. 4.11.. Example of our method vs. hg3d on UWHHI “basketball”. (a) 2D estimates of hg3d. (b) 2D reprojection of our method. (c) 3D estimates of hg3d. (d) 3D estimates of our method.

TABLE 4.3

UWHHI. AVERAGE 3D JOINT ERRORS IN MM.

Video	Person	Ours	SMPLify [20]	Hg3d
Shake Hands	S0 (green hoody)	**76.5**	116.0	128.9
	S1 (yellow hoody)	**92.1**	143.1	162.9
	S2 (white shirt)	**62.6**	140.1	N/A
Basketball	S0 (black tank)	**112.2**	162.9	127.6



(a) (b)

(c) (d)

Fig. 4.12. Example of our method vs. hg3d on UWHHI "shake hands". (a) 2D estimates of hg3d. (b) 2D reprojection of our method. (c) 3D estimates of hg3d. (d) 3D estimates of our method.

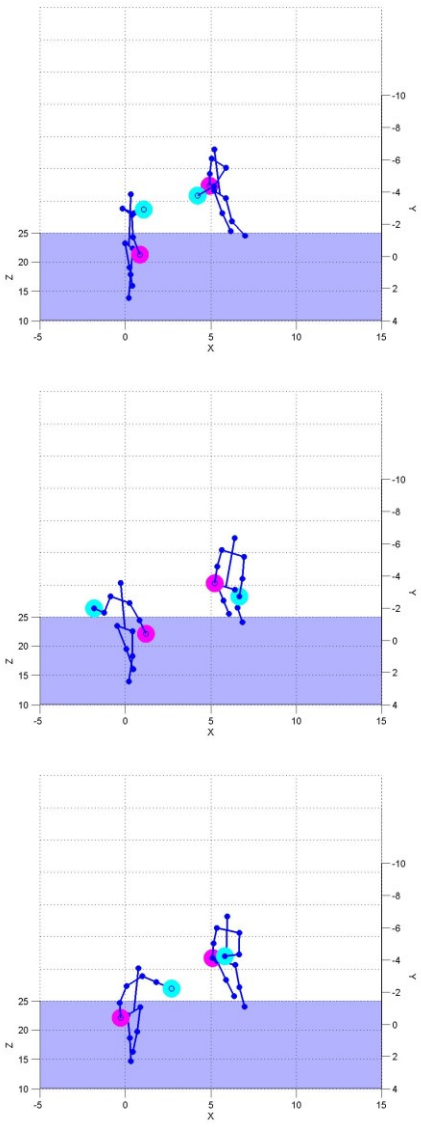
b) UCLA HHOI

We also test our 3D pose estimation performance on UCLA Human Human Interaction (HHOI) dataset [46]. Table 4.4 shows comparison with SMPLify [20] and Xiao's methods [47]. Here, s stands for skeleton-LSTM and p stands for patch-LSTM. Our method outperforms the others on this dataset by a large margin. Some qualitative results are shown in Fig. 4.13.

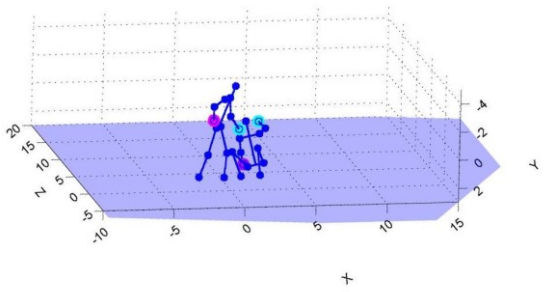
Note that our method is not directly compared with the deep learning results of this dataset reported in [46], because [46] uses the same dataset for training, while our method doesn't. None of the methods listed in Table 4.4 uses training data in UCLA HHOI dataset.

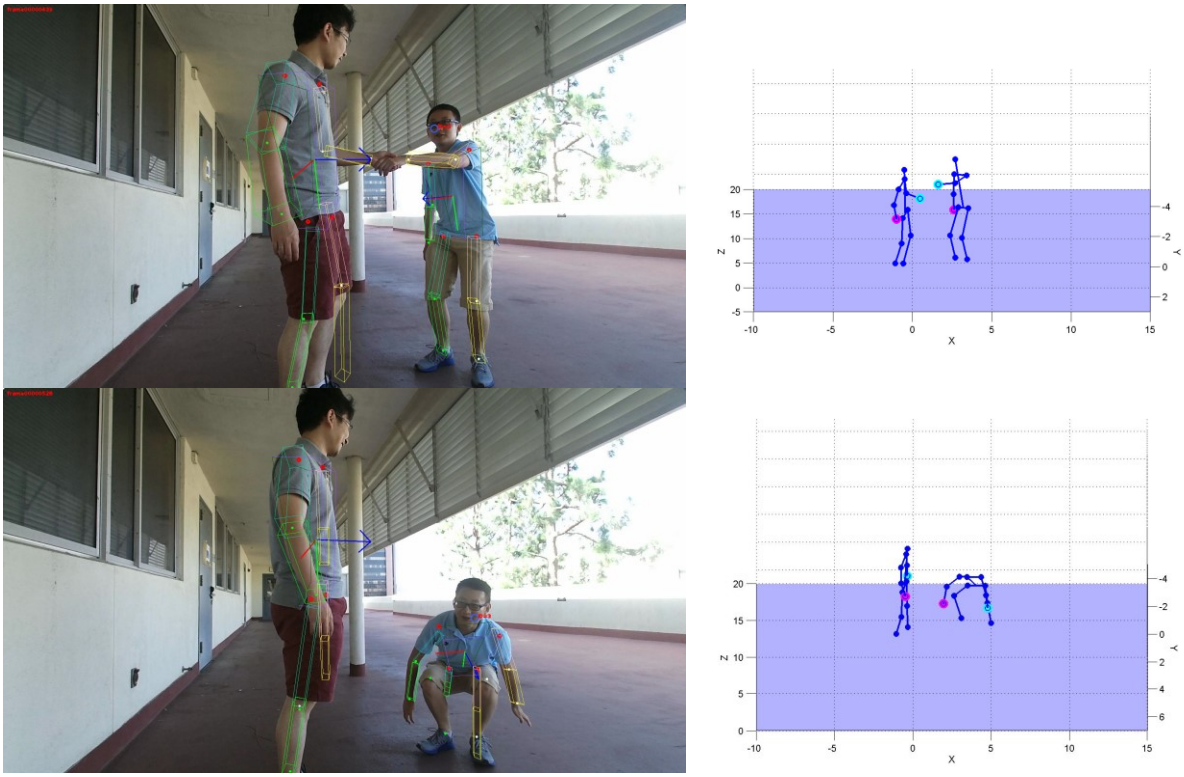
TABLE 4.4
UCLA HHOI. AVERAGE 3D JOINT ERRORS IN MM. * INDICATES RESULTS ARE OBTAINED FROM THE ORIGINAL PAPER.

Video	Ours	SMPLify [20]	Xiao [47] (s+p)	Xiao [47] (p)	Xiao [47] (s)
Hand Over	**92.9**	136.0	101.9	102.5	105.2
Pull Up	**115.6**	154.6	124.8	132.4	139.8
Shake Hands	**80.9**	122.1	118.6	129.0	113.1
High Five	**90.6**	135.6	96.1	103.0	98.4

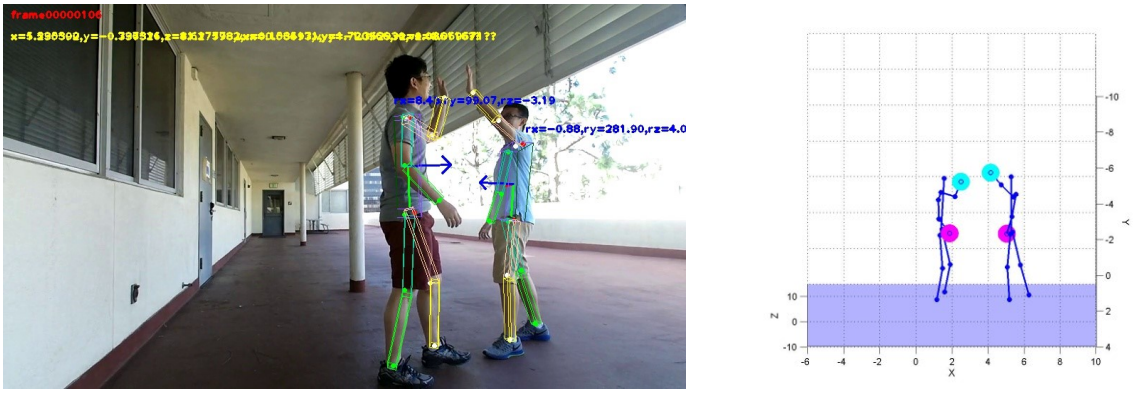


(a) Hand Over





(b) Pull Up



(c) High Five

(d) Shake Hands

Fig. 4.13. UCLA HHOI dataset results

c) Human3.6M

Our algorithm is targeted at moving camera in uncontrolled environments. However, to the best of our knowledge, there is no public dataset of such kind with 3D ground truth available. Therefore, we validate our method on Human3.6M, recorded by static cameras. The Human3.6M dataset contains 3.6M human poses from actors. The videos are captured in a controlled environment from 4 different static cameras while accurate 3D poses are measured using a MoCap system.

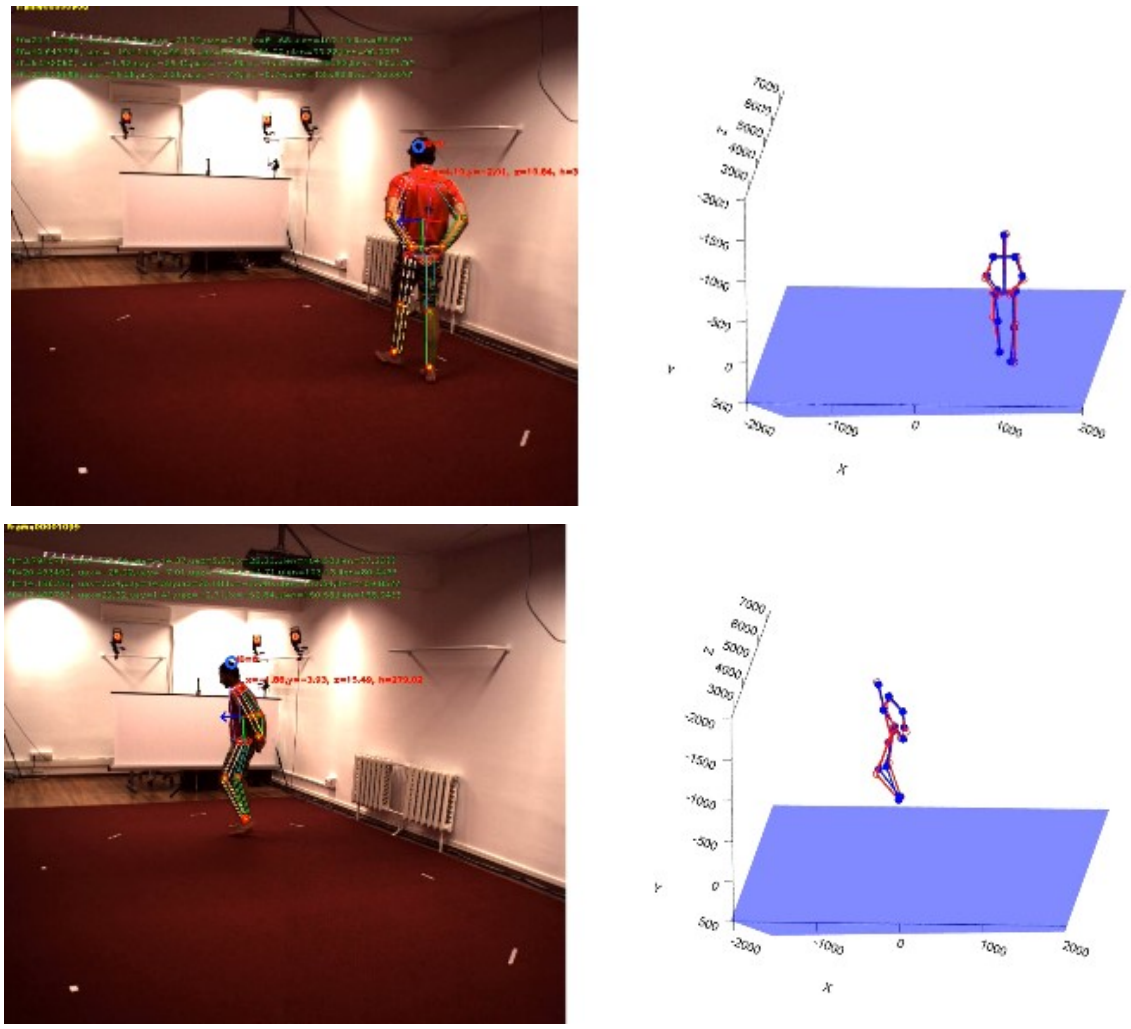


Fig. 4.14. Examples of 3D human pose estimation on Human3.6M. Red skeletons are ground truth from MoCap. Blue skeletons are estimated poses.

We validated our results on Human3.6M using the protocol as [20], where frames from subjects S9 and S11 are used for testing. Table 4.5 shows that our method outperforms state-of-the-art (2018) method [24] on almost all of the actions listed, where the best performance is shown in black bold font, while the second best is shown in blue bold font. Our method achieves quite rivaling performance as SMPLify [20], despite that SMPLify trains a regressor from the SMPL body shape to the 3D joint representation used in the dataset,

while we do not use any training data. Moreover, SMPLify does not consider multiple people and will not be able to handle occlusion caused by multiple people. SMPLify only considers single frame instead of video sequences. Besides that, due to the heavy fitting process of SMPLify, optimization for a single image takes about 1 minute on a common desktop CPU, while our method is significantly faster, i.e., our proposed method can run at 30 fps on an i5 laptop. More importantly, the performance on UCLA HHOI dataset suggests that SMPLify does not generalize well on natural videos.

Overall, our method performs best among the 2D to 3D methods on 'walk together', 'Posing', 'Waiting', 'Greeting' and 'Posing'. Our method shows lower accuracy than [24] on 'walking' action possibly because [24] model poses as a set of bases which is periodic. On contrary, we decide not to include any constraints of periodicity, and our method would be more generalized on data that is non-periodic. Some visualizations are shown in Fig. 4.14.

TABLE 4.5

QUANTITATIVE RESULTS ON HUMAN3.6M. ERRORS ARE IN MM.

Method (2D to 3D)	Pose Class										
	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk	Walk Together
Akhter & Black[36]	199.2	177.6	197.8	176.2	186.5	195.4	167.3	160.7	181.9	198.6	192.7
Ramakrishna[22]	137.4	149.3	154.3	157.7	158.9	141.8	158.1	168.6	161.7	174.8	150.2
Zhou[37]	99.7	95.8	116.8	108.3	107.3	93.5	95.3	109.1	102.2	110.4	115.2
SMPLify[20]	**62.0**	**60.2**	76.5	**92.1**	**77.0**	73.0	**75.3**	**100.3**	77.3	86.8	81.7
Wang[24]	90.3	117.6	111.0	123.5	154.9	100.5	97.3	130.6	110.3	**65.0**	88.0
Ours	75.9	94.6	**75.2**	106	99	**72.3**	94.4	106	**76.3**	75.3	**78.6**

4.5.4 *Ablation Study*

a) Hierarchical vs. Non-Hierarchical

We also investigate the impact of the hierarchical design of our method. As a control group, we disable the hierarchical estimation and optimize for 13 joints all at once. We experimented on subset of Human3.6M dataset. The results are shown in Table 4.6. We also experimented on HHOI dataset. The results are shown in Table 4.7. Our hierarchical method outperforms the non-hierarchical version. Moreover, it drastically increases computation efficiency. This advantage makes our hierarchical method promising for real-time applications.

TABLE 4.6

ABLATION STUDY: IMPACT OF HIERARCHICAL DESIGN AND OCCLUSION REASONING ON HUMAN3.6M. ERRORS IN MM.

	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk	Walk T
Non Hierarchical	79.6	98.6	83.4	107.9	109.2	74.7	103.3	119.4	83.9	91.5	91.2
W/O occlusion reasoning	77.5	90.4	77.2	96.9	95	75	96.8	104.1	76.1	80.3	78.9
Hierarchical +occlusion reasoning	**73.9**	**86.6**	**73.8**	**95.4**	**92.7**	**71**	**93.8**	**103.4**	**71.7**	**75.5**	**74.7**

TABLE 4.7

ABLATION STUDY. IMPACT OF HIERARCHICAL DESIGN AND OCCLUSION REASONING ON UCLA HHOI DATASET. ERRORS IN MM.

	Hand Over	High Five	Pull Up	Shake Hands
Non hierarchical	95.1	102.9	119.9	87.1
W/O occlusion reasoning	92.3	101.6	118	80.3
Hierarchical +Occlusion Reasoning	**88.8**	**94.2**	**113.9**	**77.4**

b) Varying the Parameters

We investigate the impact of varying the parameters in Eq. (4.21). We disable the second term, the third term and the fourth term λ_1 , λ_2 and λ_3 , which correspond to angle constraint (AC), bone length constraint (BLC) and temporal constraint (TC) respectively. The results on Human3.6M are summarized in Table 4.8, and the results on HHOI are summarized in Table 4.9.

TABLE 4.8

ABLATION STUDY: ON VARYING THE PARAMETERS. HUMAN3.6M. ERROR IN MM.

.

	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk	Walk T
Baseline (No constraint)	111.8	162.5	121.8	117.8	125.8	119.2	156.5	144.4	137.9	113.9	113.6
Baseline+BLC+TC (W/O angle constraint)	83.4	91.9	78.8	102.5	95.9	75.9	110.1	112.9	76.3	84.7	85.4
Baseline+AC+TC (W/O bone length constraint)	101.33	151.5	109.3	99.8	104.7	104.2	148.3	112.6	118.4	101.3	105.1
Baseline+AC+BLC (W/O temporal constraint)	77.5	90.4	77.2	96.9	95	75	96.8	104.1	76.1	80.3	78.9
Baseline+AC+BLC+ TC	**73.9**	**86.6**	**73.8**	**95.4**	**92.7**	**71**	**93.8**	**103.4**	**71.7**	**75.5**	**74.7**

TABLE 4.9

ABLATION STUDY: VARYING THE PARAMETERS. UCLA HHOI DATASET. ERRORS IN MM.

	Hand Over	High Five	Pull Up	Shake Hands
Baseline (No constraint)	101.5	97.6	139.3	99.7
Baseline+BLC+TC (W/O angle constraint)	91.9	96.8	119.2	83
Baseline+AC+TC (W/O bone length constraint)	95.5	105.7	126.3	83
Baseline+AC+BLC (W/O temporal constraint)	89.7	98	117.6	89.9
Baseline+AC+BLC+TC	**88.8**	**94.2**	**113.9**	**77.4**

c) Occlusion Handling

We also show in Table 4.6 and Table 4.7 the results w and w/o the occlusion handling strategy. The occlusion handling strategy effectively increased accuracy by a large margin.

4.6. Discussion

The proposed hierarchical 3D human pose estimation (H3DHPE) method is effective and efficient for scenarios recorded by monocular camera on the street and in the wild. It utilizes the recent advances in 2D body joints predictions as an intermediate step, associate individuals across frames to exploit each individual's temporal information. With a human body prior, we formulate the 3D human pose estimation problem hierarchically and efficiently solve the problem in a hierarchical fashion. We first formulate the torso estimation as a PNP problem and provide a highly efficient solution. Then we dissect pose estimation for each limb, formulate and solve an optimization problem such that each limb rests in a low dimensional pose space and does not interfere with each other. Experiments show that our method and natural results in real world videos. We also validate our results on walking pose in a well-received dataset recorded in constrained environment, and show it outperforms several state-of-the-art methods. This real-time CPU solution to address the challenge provides great new opportunities to understand and predict human behaviors in natural videos.

On the other hand, some issues remain in this work. First, H3DHPE estimates the human torso as a rigid coplanar trapezoid, which works as a rough model, but does not describe real-world human torso perfectly, as human can twist the body. The inaccurate estimates of torso pose will propagate to subsequent limb pose estimation. In the worst cases, it could result in optimization failure of limb poses. Second, when the torso keypoints are missing due to occlusion, H3DHPE can only use temporal information to predict the current

frame's torso pose, which is likely inaccurate in many cases. Inevitably, this will deteriorate limb pose estimation of the current frame even if limbs are not occluded. Last but not least, H3DHPE shows promising results in natural videos that contain multiple people recorded by a static or freely moving camera, but we still need more qualitative evaluation for the claimed scenarios. The results on UWHHI, a dataset recorded by a Kinect device, proves the algorithm promising but Kinect ground truth is not that accurate compared to the Motion Capture (MoCap) system used in other datasets such as Human3.6M [5]. Also, UWHHI is of small scale, only containing a few sequences.

To address the above issues, we propose two solutions. One is to propose a universal hierarchical 3D pose estimation that can deal with occluded or unreliable torso. The second is to exploit temporal information as much as we can. In this section, aUniversal Hierarchical 3D Pose Estimation (UH3DHPE) method will be described. The goal of this scheme is to wisely choose the hierarchy during optimization. In addition, we collected a new dataset, named Moving camera Multi-Human interactions (MMHuman) dataset, to validate the methods under multi-person scenes recorded by moving camera videos.

5. UNIVERSAL HIERARCHICAL 3D HUMAN POSE ESTIMATION

5.1. Universal Hierarchical 3D Pose Estimation

The advantages and the disadvantages of H3DHPE method in Chapter 3 has been discussed in Section 3.5. Algorithm-wise, H3DHPE has the following drawback: if any of the torso points is missing or unreliable, PnP method is not applicable or unreliable for this frame in the first place. The PnP method, which uses 4 torso points, is sensitive to noise in 2D detection.

In this section, we propose a solution, Universal Hierarchical 3D Pose Estimation (UH3DHPE) that can deal with occluded or unreliable torso. UH3DHPE is designed based on previously proposed H3DHPE, and compensate for the disadvantages of H3DHPE.

5.1.1 *System Overview*

An overall flowchart is shown in Fig 5.1. The 2D keypoints are first extracted followed by tracking, and sent to perform the Universal Hierarchical 3DHPE. The Universal Hierarchical 3DHPE (UH3DHPE) method contains 2 parallel processes, the first process performs the Hierarchical 3DHPE (H3DHPE), which consists of torso estimation and limb estimation, and the second process contains bottom-up direct limb estimation and torso refinement.

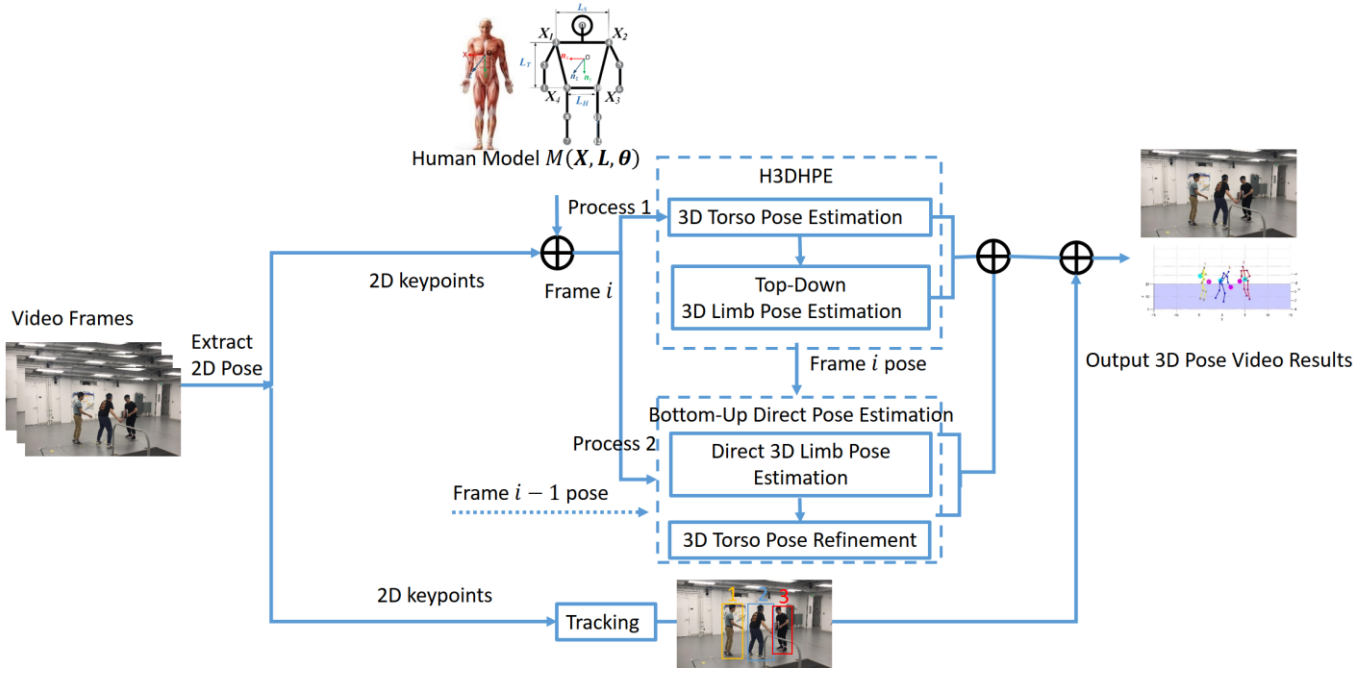


Fig. 5.1. Overall flowchart of Universal Hierarchical 3D Pose Estimation.

We inherit the setting of the system and notations from H3DHPE. The notations we are concerned about in this section are listed again, where $(\cdot)_t$ represents the corresponding variable at time t. (C) stands for Camera Coordinate, and (H) stands for Human Coordinate. In Chapter we will focus on the human pose estimation part.

Table 5.1
Notations of the System

Symbol	Notations
$\boldsymbol{R}^{(H)}$	Rotation matrix of the human pose, 3×3
$\boldsymbol{t}^{(H)}$	Translation vector of the human pose, 3×1
$\boldsymbol{K}$	Intrinsic camera matrix, 3×3
$\boldsymbol{X}^{(H)}$	3D joint in the human model coordinates, (X, Y, Z)
$\boldsymbol{X}^{(C)}$	3D joint in the camera coordinate, (X, Y, Z)
$\boldsymbol{x}$	2D joint on the image plane, $(x, y, 1)$
$l_{i,j}$	bone length in Camera Coordinate between connected joints i, j

Again, to estimate reasonable 3D poses for human, we adopt a flexible hierarchical 3D human body model. The human body model is defined as $M(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{\theta})$, parameterized by joints $\boldsymbol{X}$, bone lengths $\boldsymbol{L}$, and angles $\boldsymbol{\theta}$. Assuming known scale from Visual Odometry (See Section 3.1), we re-define the bone length between each pair of joints in Camera Coordinate as,

$$\boldsymbol{L}(i,j) = l_{i,j}, \tag{5.1}$$

where $l_{i,j}$ is the bone length between connected joints $\boldsymbol{X}_i^{(C)}, \boldsymbol{X}_i^{(C)}$, which is the same as the bone length between connected joints $\boldsymbol{X}_i^{(H)}, \boldsymbol{X}_i^{(H)}$.

In the human model coordinate system, the origin is defined as the center of the torso plane, which also determines the 3D locations of shoulder and hip joints. Angle constraints are summarized in Section 3.2.1.

5.1.2 *Revisiting Hierarchical 3D Human Pose Estimation (H3DHPE)*

In Chapter 3, a top down hierarchical method is reported. As the first step, a PNP method is adopted to estimate torso pose, $\boldsymbol{R}_k^{(H)}$and $\boldsymbol{t}_k^{(H)}$, based on the rigid body constraint. Limb points will be estimated after torso pose is resolved. From now on, we discuss per person, and do not include subscript k to denote k-th person anymore.

Given 4 pairs of joints $\{\boldsymbol{X}_i^{(C)}\}$, the torso pose $\left[\boldsymbol{R}^{(H)}|\boldsymbol{t}^{(H)}\right]$ in Camera Coordinate can be represented by:

$$\boldsymbol{X}_i^{(C)} = \boldsymbol{R}^{(H)}\boldsymbol{X}_i^{(H)} + \boldsymbol{t}^{(H)}. \tag{5.2}$$

To estimate torso pose, the following equation should be satisfied for the 4 torso points:

$$s\boldsymbol{x}_i = \boldsymbol{K}\left(\boldsymbol{R}^{(H)}\boldsymbol{X}_i^{(H)} + \boldsymbol{t}^{(H)}\right). \tag{5.3}$$

For the i-th joints of the k-th person, $\boldsymbol{X}_i^{(H)}$ is the 3D position in Human Coordinate, denoted as $\boldsymbol{x}_i = [x_i, y_i, 1]^T$, where $[x_i, y_i]$ is a 2D keypoint.

The torso pose $\left[\boldsymbol{R}^{(H)}|\boldsymbol{t}^{(H)}\right]$ can be used to transform between 3D joints in Human Coordinate $\boldsymbol{X}_i^{(H)}$ and 3D joints in Camera Coordinate $\boldsymbol{X}_i^{(C)}$.

After torso pose estimation, limb poses can then be estimated by minimizing reprojection error, regularized by angle constraints, and bone length constraints.

However, the PNP method is found to be sensitive to noise in $\boldsymbol{R}^{(H)}$. The inaccuracy of $\boldsymbol{R}^{(H)}$ can affect the subsequent limb pose estimation, which could result in optimization failure, i.e. the optimization result f does not converge to certain tolerance within a certain number of iterations N.

$$\left|f - \hat{f}\right| < \sigma\,. \tag{5.4}$$

Failure is flagged for each body part $l \in \{Right\ Arm, Left\ Arm, Right\ leg, Left\ leg, torso\}$:

$$F_l = \begin{cases} 1 & \text{optimization failure,} \\ 0 & \text{O.W.} \end{cases} \tag{5.5}$$

On the other hand, the estimation is less sensitive to noise in $\boldsymbol{t}^{(H)}$.

5.1.3 *Bottom-Up Direct Pose Estimation*

4.2.4.1. *Initialization*

There are two cases for the current frame t. First, torso points are detected, no matter how reliable it is. In this case, we perform H3DHPE in Process 1, and initialize the pose using current frame's torso pose $\left[\boldsymbol{R}^{(H)}|\boldsymbol{t}^{(H)}\right]$ in Process 2. Second, torso points are missing and Process 1 is invalid. In Process 2, we initialize the pose using torso pose of last frame t-1, or the nearest past frame with available torso pose.

4.2.4.1. *Bottom-Up Direct Pose Estimation*

To overcome the high error sensitivity of the 3D pose estimation on the $\boldsymbol{R}^{(H)}$estimated for the torso, due to unreliable or missing detections of torso 2D keypoints, in the proposed Universal Hierarchical 3D Human Pose Estimation (UH3DHPE) method, we present a bottom-up direct pose estimation method, where limb poses are first directly estimated by solving closed form equations. While U3DHPE may suffer from optimization failure as shown in left figure of Fig. 5.3 (a), the direct limb pose estimation method provides a closed form solution that is visually in accordance with the image. The input of direct limb pose estimation

for one limb (such as left leg) is the 2D keypoints on the limb $\boldsymbol{x}_i$ ($i \in \{l\}$) and the output is the joints' 3D positions in Camera Coordinate $\boldsymbol{X}_i^{(C)}$ ($i \in \{l\}$). The input of torso refinement is the initialized torso points $\boldsymbol{X}_i^{(C)}(i \in \{torso\})$, and the output is the optimized torso points.

a) Limb Pose Estimation

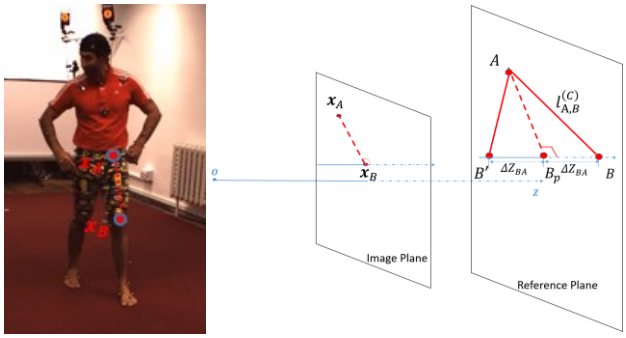


Fig. 5.2. Illustration of projected length, bone length, and depth.

For a pinhole camera, we can map the 3D joints in Camera Coordinate to image plane using the following projections**:**

$$x = \frac{f}{Z}X + c_x,$$
$$y = \frac{f}{Z}Y + c_y. \tag{5.6}$$

For simplicity, take one person's left leg for example. Consider left hip as the reference point, which is denoted as $\boldsymbol{X_A} = (X_A, Y_A, Z_A)$ in Camera Coordinate corresponding to A in Fig. 5.2, and the left knee is denoted as $\boldsymbol{X_B} = (X_B, Y_B, Z_B)$ in Camera Coordinate corresponding to B in Fig. 5.2. For notational simplicity we do not write superscript (C) in the following derivation. Given 2D keypoints $\boldsymbol{x}_A = (x_A, y_A)$, $\boldsymbol{x}_B = (x_B, y_B)$, our goal is to estimate $\boldsymbol{X_B}$ relative to $\boldsymbol{X_A}$ in 3D coordinate.

Given the bone length $l_{i,j}$, and the bone's projections on 2D, when "lifting" 2D to 3D, there is still an ambiguity. Consider a reference plane that is parallel to the image plane, and is located at the depth of joint $\boldsymbol{X_A}$, as shown in Fig. 3, the same projection B_p on reference plane could correspond to a 3D point $\boldsymbol{X_B}$ closer to camera than $\boldsymbol{X_A}$, or a 3D point $\boldsymbol{X_{B'}}$ farther from camera than $\boldsymbol{X_A}$. The length of left upper leg is denoted as $l_{A,B}$, which corresponds to AB in Fig 5.2.

Here we provide a closed form solution that only depends on torso pose $\boldsymbol{t}^{(H)}$ (distance of human to camera), but not $\boldsymbol{R}^{(H)}$. More specifically,

X_A, Y_A, X_B, Y_B in Camera Coordinate are:

$$X_A \approx \frac{Z}{f}(x_A - c_x), Y_A \approx \frac{Z}{f}(y_A - c_y),$$
$$X_B \approx \frac{Z}{f}(x_B - c_x), Y_B \approx \frac{Z}{f}(y_B - c_y), \tag{5.7}$$

where depth Z is approximated using initialized torso translation $\boldsymbol{t}^{(H)}$along Z axis, which is the torso center's depth, or the human's depth. The approximation is valid because the depth variation of the human himself/herself is considered small compared to depth to the camera. This approximation basically finds the scale $\frac{Z}{f}$ that maps human in pixel space to Camera Coordinate.

The length of the bone's projection on the reference plane is denoted as AB_p, as shown in Fig 3. Given$B_p = (X_B, Y_B, Z)$, we can calculate the projected length AB_p by:

$$AB_p = \sqrt{(X_A - X_B)^2 + (X_A - X_B)^2} = \frac{Z}{f}\|\boldsymbol{x}_A - \boldsymbol{x}_B\|, \tag{5.8}$$

AB_p only depends on initialized torso translation $\boldsymbol{t}^{(H)}$, but not rotation $\boldsymbol{R}^{(H)}$.

According to geometry we have:

$$BB_p{}^2 + AB_p{}^2 = AB^2, \tag{5.9}$$

where $AB = l_{A,B}$, AB_p is calculated from Equation (5.8). $BB_p = |Z_B - Z_A|$ is the depth difference to be calculated.

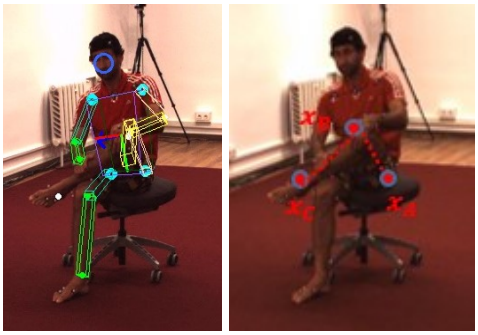
Thus we have:

$$(Z_B - Z_A)^2 + (\frac{Z}{f}\|\boldsymbol{x}_A - \boldsymbol{x}_B\|)^2 = l_{A,B}{}^2. \tag{5.10}$$

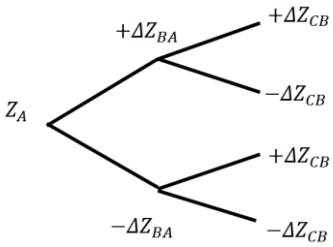
Therefore, we can calculate the relative depth:

$$Z_B - Z_A = \pm\sqrt{l_{A,B}{}^2 - (\frac{Z}{f}\|\boldsymbol{x}_A - \boldsymbol{x}_B\|)^2}. \tag{5.11}$$

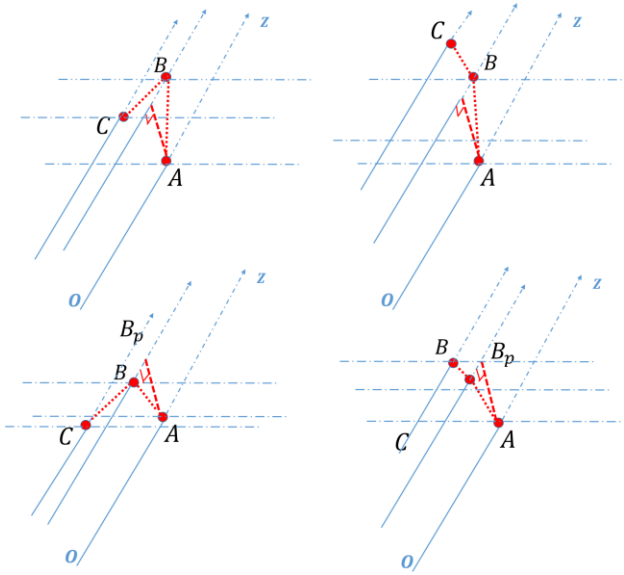
Based on (5.11), there are 2 possible solutions for Z_B given Z_A. For each limb, there is an upper limb and a lower limb. As shown in Fig. 5.3, therefore we have 4 possible solutions for each limb, given the root joint of a limb such as left hip. The solution set for left leg is denoted as S_{LL}. Similarly we have S_{RL} for right leg, S_{RA} for right arm, S_{LA} for left arm.



(a) Optimization fails for left leg in H3DHPE. We use 2D keypoints on left leg to directly estimate leg pose.



(b) A tree of possible solutions of a limb given a root joint



(c) Visualization of 4 possible solutions.

Fig. 5.3. To handle optimization failure cases, directly estimate limbs based on keypoints.

b) Removing redundant solutions

We can use the following two constraints to remove redundant solutions.

Angle constraints: Since $\Delta Z_{BA}, \Delta Z_{CB}$ are known, the 3D vectors $\boldsymbol{X_B} - \boldsymbol{X_A}$ and$\boldsymbol{X_B} - \boldsymbol{X_C}$ are in closed form, and the angle between these two vectors can be calculated by $arccos \frac{(\boldsymbol{X_B}-\boldsymbol{X_A})\cdot(\boldsymbol{X_B}-\boldsymbol{X_C})}{\|(\boldsymbol{X_B}-\boldsymbol{X_A})\|\|(\boldsymbol{X_B}-\boldsymbol{X_C})\|}$. Based on the angle constraints defined in the human model $M(\boldsymbol{X}, \boldsymbol{L}, \boldsymbol{\theta})$, we can prune the solution tree in Fig. 4 (b) if possible and only keep the solutions that satisfy angle constraints.

Temporal constraints: We apply the temporal constraints to further remove redundant solutions.

The cost function of the joints on any limb is defined as follows,

$$f_{\boldsymbol{T}}\left(\boldsymbol{X}_{LL}^{(C)}\right)=\sum_{i\epsilon \boldsymbol{LL}}\left\|\boldsymbol{X}_{i,t}^{(C)}-\boldsymbol{X}_{i,t-1}^{(C)}\right\|^{2}, \tag{5.12}$$

where $\boldsymbol{X}_{LL}^{(C)}$ is the joints on the left leg of the k-th person

We find the solution $\boldsymbol{X}_{LL}^{(C)}\epsilon S_{LL}$ should be the one that minimizes $f_{\boldsymbol{T}}\left(\boldsymbol{X}_{LL}^{(C)}\right)$. In other words, we find the solution that is closest to last valid frame's pose. After this step, we only have one closed form solution, denoted as $\Delta\tilde{Z}_{BA}$ and $\tilde{Z}_{CB}$, then the solution of the 3D limb can be obtained by:

$$\begin{aligned}\boldsymbol{X_B}&=\left(\frac{Z}{f}(x_B-c_x),\frac{Z}{f}\left(y_B-c_y\right),\mathrm{Z}+\Delta\tilde{Z}_{BA}\right),\\ \boldsymbol{X_C}&=\left(\frac{Z}{f}(x_C-c_x),\frac{Z}{f}\left(y_C-c_y\right),\mathrm{Z}+\Delta\tilde{Z}_{BA}+\Delta\tilde{Z}_{CB}\right).\end{aligned} \tag{5.13}$$

Same as in Equation (5.7), depth Z is still approximated using the initialized torso translation $\boldsymbol{t}^{(H)}$along Z axis, and can be refined later.

c) Torso Pose Refinement

In this step, we refine the torso pose $\boldsymbol{R}^{(H)}$ by refining the three angles $\boldsymbol{R}_X^{(H)},\boldsymbol{R}_Y^{(H)},\boldsymbol{R}_Z^{(H)}$ that form the Rodrigues vectors of $\boldsymbol{R}^{(H)}$. In addition, we refine the t_z value in $\boldsymbol{t}^{(H)}=[t_x,t_y,t_z]^T$. The cost function is derived as:

$$\begin{aligned}f\left(\boldsymbol{X}_i^{(C)}\,|\boldsymbol{R}_X^{(H)},\boldsymbol{R}_Y^{(H)},\boldsymbol{R}_Z^{(H)},\boldsymbol{t}^{(H)}\right)&=\sum_i\left\|\boldsymbol{K}\boldsymbol{X}_i^{(C)}-s\boldsymbol{x}_i\right\|,\\ s.t.\left\|\boldsymbol{X}_i^{(C)}-\boldsymbol{X}_j^{(C)}\right\|&=l_{i,j}^{(C)}\ ,\end{aligned} \tag{5.14}$$

where $\{\boldsymbol{X}_i^{(C)}\}$ is the un-occluded joints in the set of torso points. s is the scale factor, determined by $\frac{Z}{f}$.

The solved torso pose might not be perfect, but it will not cause limb estimation to fail, thus having less influence on the limb estimation.

The proposed bottom-up limb pose method still allows to dissect the calculation and allow parallel processing efficiently, i.e., each limb can be estimated in parallel without mutual dependency. For multiple persons, each person can be processed in parallel as well.

d) Adaptive Weights Merging Strategy (AWMS) for Concurrent Processes

After obtaining the pose results using the newly proposed bottom-up limb estimation and torso refinement, i.e., the Bottom-Up Direct Pose Estimation method, we design a strategy to merge the results with the original H3DHPE. The reason we design a weighted merging strategy for the two concurrent processes is: first, the proposed bottom-up algorithm is based on the assumption that the camera's distance is much larger than the human limb length itself. This is an assumption that original H3DHPE does not require. Therefore, there could be certain cases where the original H3DHPE works better. Second, as mentioned before, the inaccuracy of $\boldsymbol{R}^{(H)}$ in H3DHPE can affect the subsequent limb pose estimation. In such cases, we should adopt the newly proposed Bottom-Up Direct Pose Estimation method described in Section 5.1.3 a), or it should have higher weight.

The weight for prioritizing H3DHPE or the proposed new algorithm is calculated adaptively depending on optimization results flagged by (4.5):

$$\boldsymbol{X}_{UH}^{(C)}(l) = (1 - F_l)\lambda\, \boldsymbol{X}_{p1}^{(C)}(l) + \frac{(1-\lambda)}{(1-\lambda) + (1 - F_l)\lambda} \boldsymbol{X}_{p2}^{(C)}(l), \tag{5.15}$$

where $\boldsymbol{X}_{p1}^{(C)}$ represents estimated results from H3DHPE (Process 1) and $\boldsymbol{X}_{p2}^{(C)}$ represents the new proposed bottom-up solution (Process 2).

Case 1. If all limbs are marked success, $F_l = 0$, it becomes:

$$\boldsymbol{X}_{UH}^{(C)}(l) = \lambda\, \boldsymbol{X}_{p1}^{(C)}(l) + (1-\lambda)\boldsymbol{X}_{p2}^{(C)}(l), \tag{5.16}$$

where weights λ is a fixed hyper parameter when calculating the final 3D poses $\boldsymbol{X}_{UH}^{(C)}$. This strategy is suitable when neither H3DHPE nor the new Bottom-Up Algorithm shows obvious flaws. Theoretically, when torso keypoints are relatively accurate, and not subject to occlusion, and the distance of human subjects to the camera is much larger than the bone lengths, both H3DHPE and the new Bottom-Up Algorithm are effective.

Case 2. In the cases that $F_l = 1$, for body part l:

$$\boldsymbol{X}_{UH}^{(C)}(l) = \boldsymbol{X}_{p2}^{(C)}(l)\,, \tag{5.17}$$

i.e., the Bottom-Up Algorithm is adopted.

5.2. Experiments and Analysis

5.2.1 *Experimental Setting*

For quantitative evaluations, we experiment on Human3.6M [5], UWHHI [51], UCLA HHOI [46] and MMHuman20K, all of which have corresponding 3D joint ground truth labeled. A summary of all the datasets

used to perform quantitative evaluations are listed in Table 5.2. For qualitative evaluations, we also show our performance on several representative videos, and include some comparisons for videos that were given as demos in other works. Ablation study is also conducted.

All of our experiments are conducted on Windows 10, using an Intel Core i5-6300HQ CPU@2.30Ghz, 2301MHz, 4 Core Processor. We use OpenPose [1] as the 2D keypoint detector, and λ is set to 0.6 in our experiments.

TABLE 5.2
SUMMARY OF DATASETS

Dataset	UCLA HHOI [46]	Human3.6M [5]	UWHHI [51]	MMHuman20K
Moving Camera	no	no	no	yes
Multiple People	yes	no	yes	yes
Inter-person Occlusions	no	no	no	yes
Ground Truth	Kinect	MoCap	Kinect	MoCap
Num. Actions	4	11	2	4

5.2.2 *Quantitative Evaluation*

a) UWHHI

As shown in the comparative performance on University of Washington Human-Human Interaction (UWHHI) dataset in Table 5.3, our proposed UH3DHPE method outperforms state-of-the-art methods that reported on this dataset for multi-person human pose estimation using a moving camera in natural scenarios.

Although H3DHPE usually gives a reasonable pose, it may not be that accurate and one or more of the limb pose estimation could fail. As shown in Fig. 5.4, we can see one example, where H3DHPE gives a bad estimate of the leg, while UH3DHPE can successfully recover the 3D poses.

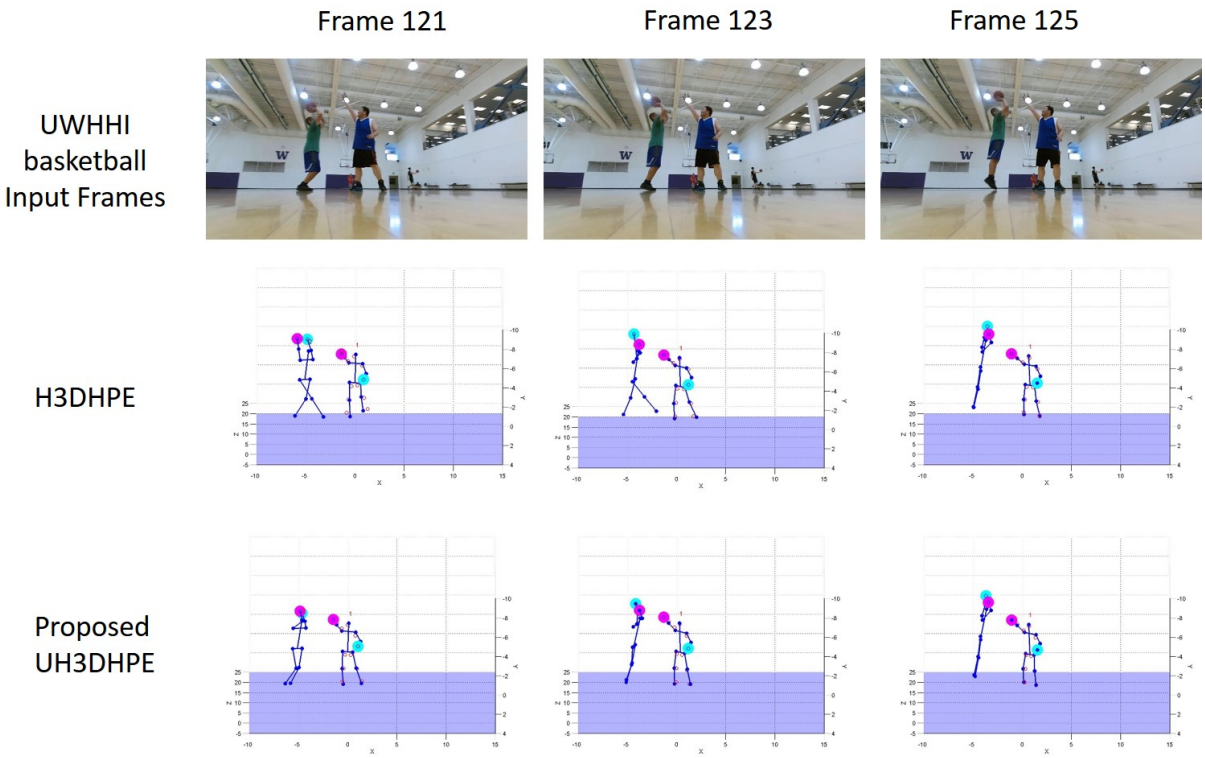


Fig. 5.4. Example of proposed method vs. [19] on UWHHI "basketball". The first row is the input frames. The 2nd row is estimation results of H3DHPE. The 3rd row is estimation results of proposed UH3DHPE.

TABLE 5.3
UWHHI. AVERAGE 3D JOINT ERRORS IN MM.

Video	Shake Hands			Basketball
	S0 (green hoody)	S1 (yellow hoody)	S2 (white shirt)	S0 (black tank)
SMPLify [20]	116.0	143.1	140.1	162.9
Hg3d [32]	128.9	162.9	N/A	127.6
H3DHPE [70]	76.5	92.1	62.6	112.2
Proposed	**72.9**	**86.6**	**52.8**	**94.9**

b) MMHuman20K

For MMHuman20K, which is not enough for training, it is fair to only use MMHuman20K as a test set and compare the results with other methods. We compare our method with several open source state-of-the-art methods. Since many state-of-the-art work provide solutions only for a single person, for the comparison methods, we input tracked 2D keypoints with missing 2D keypoints interpolated. To fairly evaluate our performance, all the comparison methods use the same 2D keypoints detected by OpenPose.

Experimental results compared with VideoPose3D [63], Hossain et al. [65], and Lin et al. [66] are reported in Table 5.4. Some qualitative comparisons are shown in Fig. 5.5.

Average 3D errors are Mean Per Joint Position Error (MPJPE) in millimeter between estimated pose and the ground-truth. Our UH3DHPE performs on par or better than state-of-the-art.

On average, our UH3DHPE outperforms state-of-the-art VideoPose3D by 7%. This suggests that even though methods trained on labelled 3D data reports a high accuracy on the test set in the same dataset like Human3.6M, they may not generalize to the complexity of other scenarios.

TABLE 5.4
QUANTITATIVE RESULTS ON **MMHUMAN20K**. ERRORS ARE IN MM.

Method (W/O training on MMHuman)	Pose Class							
Sequence ID / Avg.	Walk Cross 2503	Walk Cross 3281	Walk Cross 3282	Walk Cross Average	High Five 2505	High Five 3283	High Five 3284	High Five Avg.
Hossain et al. [65]	126.0	122.5	134.3	127.6	105.6	113.55	123.8	114.3
Lin et al. [66]	131.0	112.9	128.7	124.2	119.9	120.0	125.9	121.9
VideoPose3D[63]	92.5	81.9	94.9	89.7	**73.0**	77.6	77.6	76.0
Proposed UH3DHPE	**84.99**	**80.0**	**76.4**	**80.5**	80.3	**73.3**	**70.6**	**74.7**
Sequence ID / Avg.	Hand Over 2507	Hand Over 3287	Hand Over 3288	Hand Over Avg.	Shake Hands 2501	Shake Hands 3285	Shake Hands 3286	Shake Hands Avg.
Hossain et al. [65]	108.5	109.7	119.3	112.5	121.8	115.1	131.9	122.9
Lin et al. [66]	99.8	108.7	114.2	107.6	121.9	106.99	115.6	114.8
VideoPose3D [63]	67.1	74.2	77.7	73	91.5	71.6	**95.0**	86.0
Proposed UH3DHPE	**65.5**	**67.5**	**69.1**	**67.4**	**67.2**	**71.0**	97.0	**78.4**

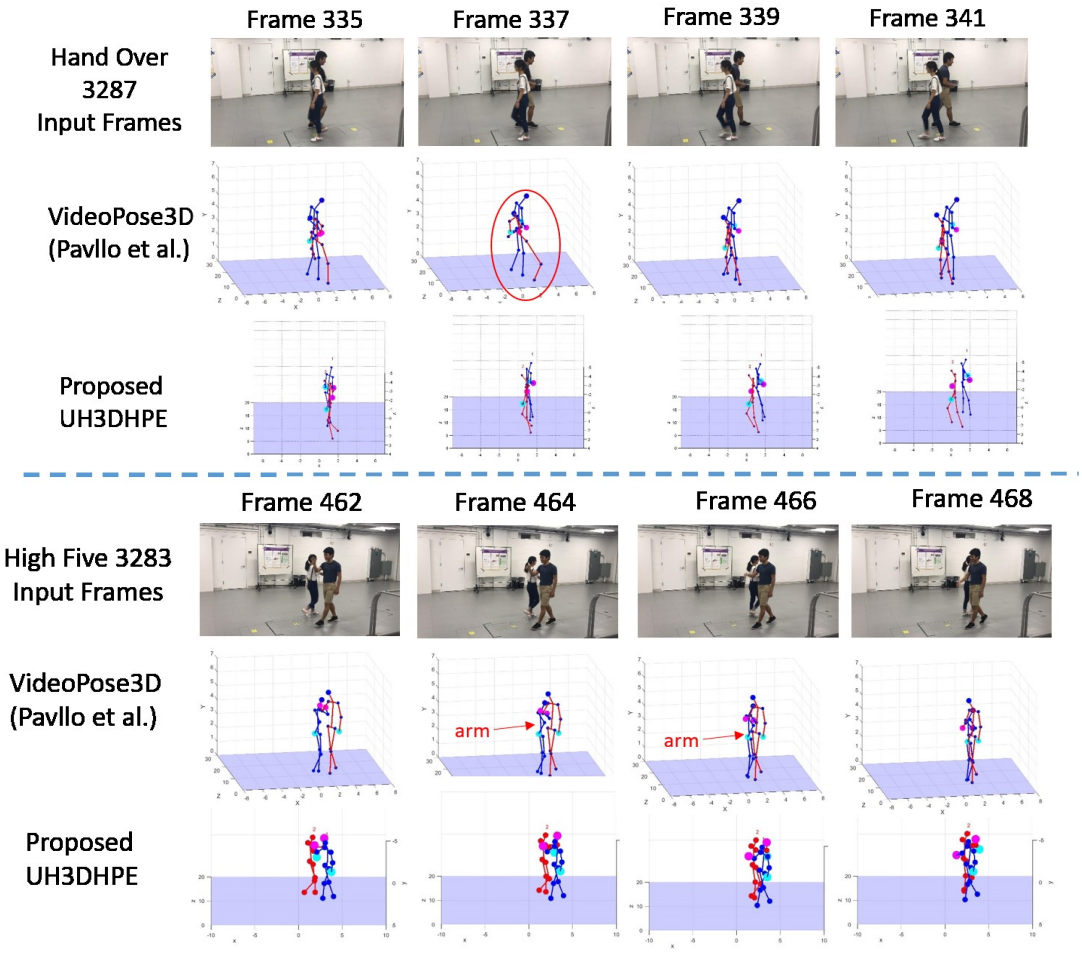


Fig. 5.5. MMHuman20K dataset results. Examples of 3D human pose estimation compared to state-of-the-art VideoPose3D [24]. The upper figure shows results of Sequence "Hand Over 3287", and the lower figure shows results of Sequence "High Five 3283".

c) UCLA HHOI

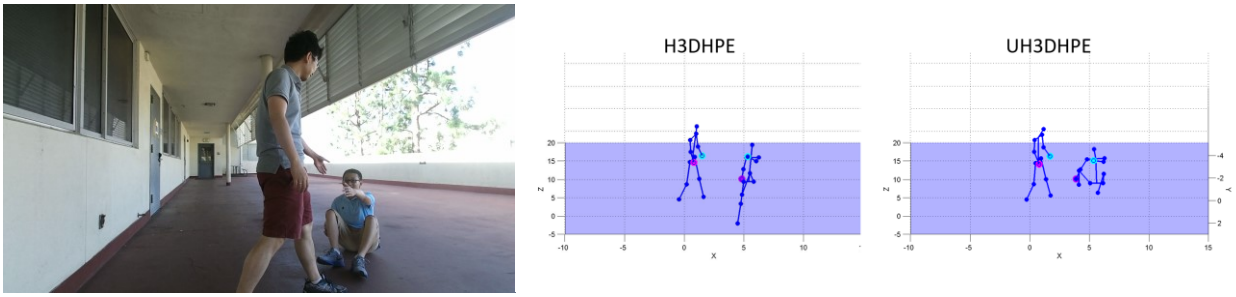
We also test our 3D pose estimation performance on the UCLA Human-Human-Object Interaction (HHOI) dataset [33]. Table 5.5 shows comparison with SMPLify [20], Xiao's methods [32], and H3DHPE [70]. In this table, *s* stands for skeleton-LSTM and *p* stands for patch-LSTM. Our method outperforms the available solutions published on this dataset. Our method outperforms H3DHPE [] on most of the classes quantitatively except "Hand Over". The proposed UH3DHPE makes biggest improvement on "Pull Up", where torso pose is the most unreliable among all classes. The slight performance degradation on "Hand Over" may be due to the fact that UCLA HHOI dataset does not have inter-person occlusions, therefore the advantage of merging an alternative limb pose estimation algorithm is mitigated and the disadvantage of approximation errors when camera is close to the subject takes place. When looking at the qualitative results, UH3DHPE gives limb estimates that are in better accordance with human vision, as shown in Fig. 5.6.

Note that our method is not directly compared with the deep learning results of this dataset reported in [33], which directly uses the same dataset for training, while our method only uses it as the test set. None of the methods listed in Table 5.5 uses UCLA HHOI dataset as the training data.

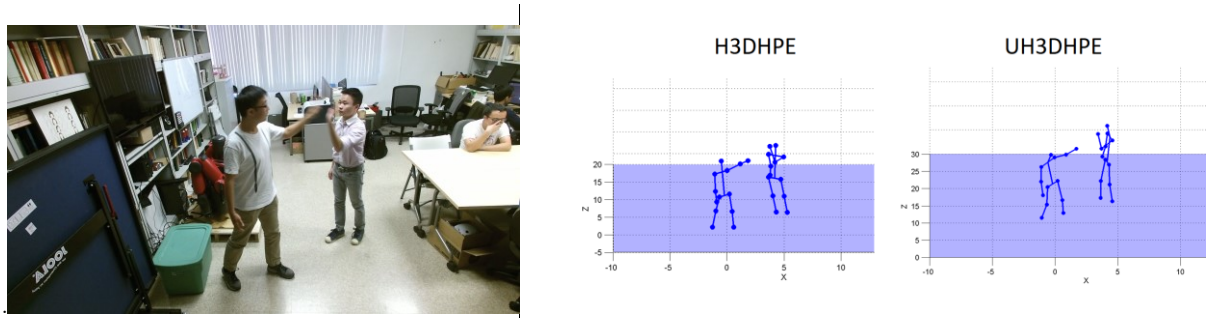
TABLE 5.5
UCLA HHOI. AVERAGE 3D JOINT ERRORS IN MM.

Video	Hand Over	Pull Up	Shake Hands	High Five
SMPLify [20]	136.0	154.6	122.1	135.6
Xiao [32] (s+p)	101.9	124.8	118.6	96.1
Xiao [32] (p)	102.5	132.4	129.0	103.0
Xiao [32] (s)	105.2	139.8	113.1	98.4
H3DHPE [70]	**88.8**	113.9	77.4	94.2
Proposed H3DHPE	90.2	**109.2**	**76.4**	**93.7**

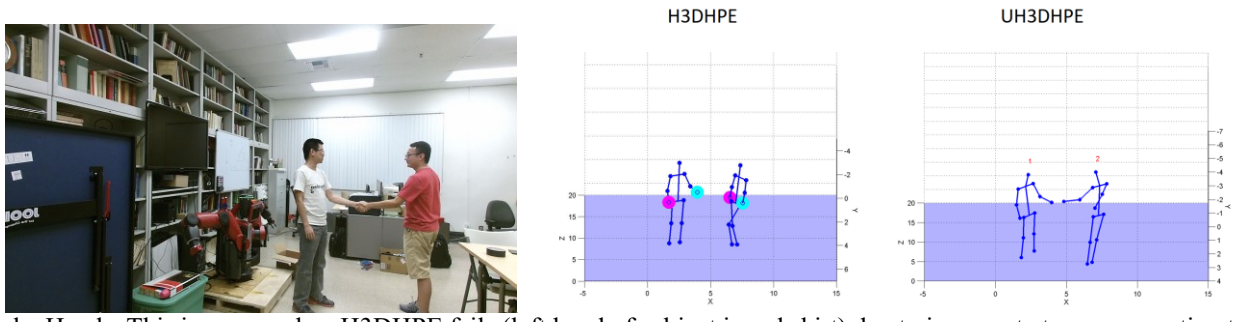
(a) Hand Over. UH3DHPE has better leg pose estimation compared to H3DHPE.



(b) Pull Up



(c) High Five. UH3DHPE has better limb estimation compared to H3DHPE.



(d) Shake Hands. This is a case when H3DHPE fails (left hand of subject in red shirt) due to inaccurate torso pose estimation. On the contrary, UH3DHPE can give correct estimations.

Fig. 5.6. UCLA HHOI dataset results

d) Human3.6M

Our pose estimation algorithm is targeted at the challenging moving camera scenarios in uncontrolled environments, especially when training using 3D training data is not available. Human3.6M contains single-person activity videos captured by a static camera, which is actually not our main focus. However, Human3.6M provides a variety of single-person actions recorded by static cameras. Therefore, we validate our method on Human3.6M, and compare our results to other methods that do not use labelled 3D training data. One small exception in the compared methods is SMPLify [20], which trains a regressor from the SMPL body shape to the 3D joint representation used in the dataset, but does not use 3D training in any other way.

We validated our results on Human3.6M using the protocol defined in [20], where frames from subjects S9 and S11 are used for testing. Table 5.6shows that our method outperforms state-of-the-art unsupervised methods as reported up to 2019 on the majority of the actions listed, where the best performance is shown in black bold font, while the second best is shown in blue bold font. Our method achieves quite comparable performance as SMPLify [20], despite that SMPLify trains a regressor from the SMPL body shape to the 3D joint representation used in the dataset, while we do not use any training data. Moreover, SMPLify does not consider multiple people and will not be able to handle occlusions caused by multiple people. SMPLify only considers single frame instead of video sequences.

TABLE 5.6
QUANTITATIVE RESULTS ON HUMAN3.6M. ERRORS ARE IN MM.

Unsupervised Two-stage Methods	Pose Class										
	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk	Walk Together
Ramakrishna[22]	137.4	149.3	154.3	157.7	158.9	141.8	158.1	168.6	161.7	174.8	150.2
Zhou[30]	99.7	95.8	116.8	108.3	107.3	93.5	95.3	109.1	102.2	110.4	115.2
SMPLify[20]	**62.0**	**60.2**	76.5	92.1	**77.0**	73.0	**75.3**	100.3	77.3	86.8	81.7
Wang[23]	90.3	117.6	111.0	123.5	154.9	100.5	97.3	130.6	110.3	**65.0**	88.0
H3DHPE [19]	75.9	94.6	75.2	106	99	72.3	94.4	106	76.3	75.3	78.6
Proposed UH3DHPE	71.9	82	**73.4**	**84.4**	86.6	**66.5**	79.4	**93.3**	**72.3**	79.2	**77.2**

5.2.3 *Qualitative Results*

a) BoyDance - YouTube

The BoyDance video, originated from YouTube, is from Kinectics-400 dataset [78]. A demo of this video is given in the slides for the Temporal3DPose method [77]. We compare our results qualitatively with Temporal3DPose [77] and HMR [60] in Fig. 5.7. Temporal3DPose shows better performance than HMR model, but still suffer from optimization failure in some cases (See Frame 149), while our UH3DHPE method consistently gives reasonable estimates.

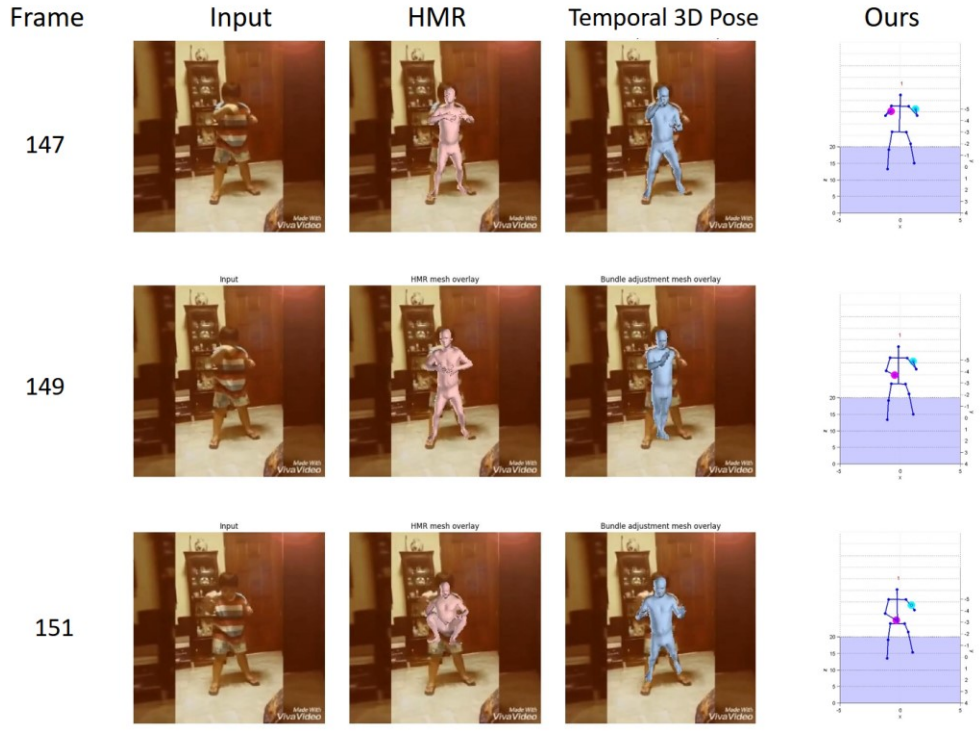


Fig. 5.7. Comparison of performance on boy dancing video.

b) Basketball (medium level)

Qualitative results on a self-recorded video where actors are playing basketball is shown in Fig. 5.8. Our solution can handle the scenario containing complicated inter-human interactions.

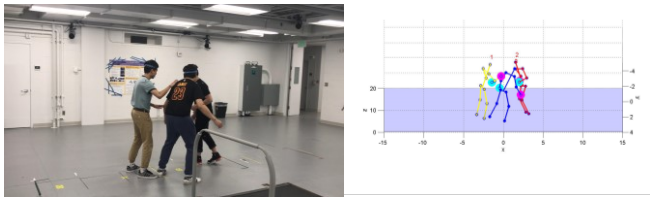


Fig. 5.8. Multi-person 3D human pose estimation for medium level basketball scenario.

c) Basketball (hard level)

Qualitative results on a self-recorded video where real players are playing basketball is shown in Fig. 5.9. In this video, there are a lot of occlusions and the scene is totally natural and players are moving in in fierce confrontations and fast motion.

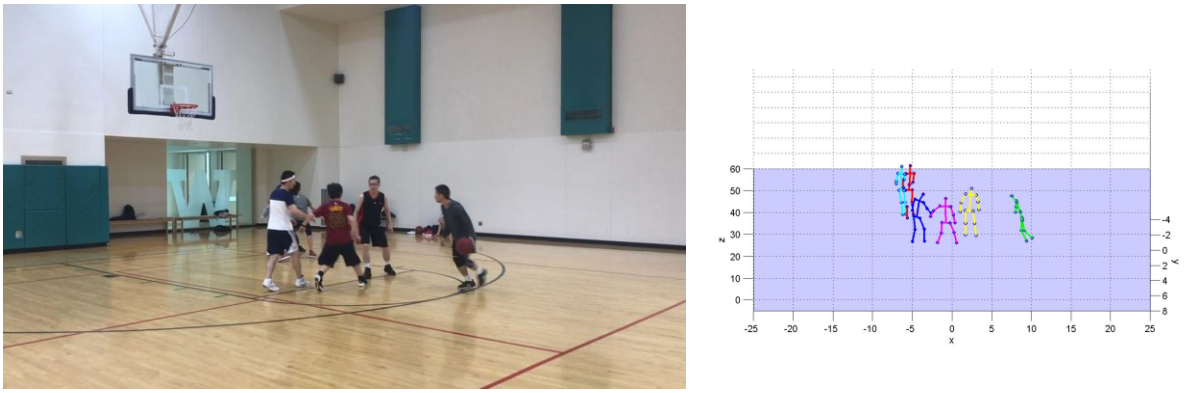


Fig. 5.9. Multi-person 3D human pose estimation for natural basketball scene.

5.2.4 *Ablation Study*

a) Adaptive Weights Merging Strategy

We compare our bottom-up method (baseline) vs. UH3DHPE (baseline+AWMS) on the Human3.6M dataset. From Table 5.7 we can see UH3DHPE works generally better than the baseline on various poses. This proves our Adaptive Weights Merging Strategy effective.

TABLE 5,7
IMPACT OF AWMS ON HUMAN3.6M. ERRORS IN MM.

	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk T	Average
baseline	76.1	93.2	80.0	**80.8**	92.9	72.9	84.1	97.0	86.5	90.8	85.4
baseline+AWMS	**71.9**	**82**	**73.4**	84.4	**86.6**	**66.5**	**79.4**	**93.3**	**72.3**	**79.2**	**77.2**

b) Varying the Parameters

We vary the parameter λ. In Table 5.8, we show the results for λ=0.4, λ=0.6 and λ=0.8, where higher λ means H3DHPE is weighted higher over the Bottom-Up Direct Pose Estimation method in our UH3DHPE framework. The dataset human3.6M for ablation study does not contain inter-person occlusions, and we

found λ=0.6 gives the best performance, meaning H3DHPE is weighted slightly higher over the Bottom-Up Direct Pose Estimation Method in UH3DHPE.

TABLE 5.8
IMPACT OF VARYING PARAMETERS ON HUMAN3.6M. ERRORS IN MM.

	Directions	Discussion	Greeting	Phoning	Photo	Posing	Purchases	Sit	Waiting	Walk T	Average
(λ =0.4)	**71.8**	87.8	76.2	**81.1**	92.9	68.8	81.8	94.4	79.7	83.2	81.8
(λ =0.6)	71.9	**82**	73.4	84.4	**86.6**	**66.5**	**79.4**	**93.3**	**72.3**	79.2	**77.2**
(λ =0.8)	71.9	85.0	**72.7**	89.8	92.9	68.0	86.1	98.3	73.3	**75.8**	81.4

5.3. Discussion

Multi-person 3D pose estimation using a monocular freely moving camera in real-world scenarios remains a challenge for real-world scenes usually contain self-occlusions and inter-person occlusions. The H3DHPE in Chapter 3 provides a handy unsupervised solution for natural videos. However, H3DHPE is sensitive to unreliable or missing torso keypoints detections. To address the challenges, we propose an unsupervised Universal Hierarchical 3D Human Pose Estimation (UH3DHPE) method. To handle the case of an occluded or inaccurate 2D torso keypoints, which play an important role for 3D pose initialization and subsequent inference, an effective method to directly estimate limb poses without building upon the estimated torso pose is proposed, and the torso pose can then be further refined to form the hierarchy in a bottom-up fashion. An adaptive merging strategy is proposed to determine the best hierarchy. To verify the effectiveness of the proposed scheme, a video dataset for multi-person interactions is collected by a moving camera, under a MoCap ground truth data acquisition environment, for our performance evaluations. Experimental results shows our proposed UH3DHPE method outperforms state-of-the-art methods, including deep learning methods trained on existing datasets, on multi-person scenes with occlusions recorded by moving camera. Our framework gives a decent estimate for challenging natural scenes such as basketball.

6. TRAJECTORY RECOVERY

As discussed in Section 3.5. The second proposed solution is to exploit temporal information as much as we can to recover trajectory. There are several goals to propose a trajectory recovery strategy. 1) Handle occlusion. In 2D, occlusion is a challenging problem. Occlusion include self-occlusion and inter-person occlusion, resulting in missing joints. 2) exploit spatial-temporal information so that motion patterns can be recovered. 3) The proposed strategy should be robust to noise.

If the window of missing joints is too large, simple methods such as interpolation may not work so well to recover the movement patterns. In this case, we seek to adopt other novel recovery methods. In this section, we propose a novel 2D to 3D trajectory recovery method that leverages the power of GAN.

There are several advantages to use a neural network for trajectory recovery. First is to solve occlusion for longer sequences. Second, give a good initial value with pretrained model. Third, reduce noise. Along time, the raw 2D trajectory often has noise. Kalman filter can solve part of the problem, but usually it is still noisy. Fourth, to make the pose estimates more natural, taking advantage of the power of GAN.

6.1. Approach

6.1.1 *Temporal Gated Convolution*

Given a temporal sequence of detected 2D joint locations $\boldsymbol{x} \in \mathbb{R}^{2N_{jt} \times T}$, our goal is to reconstruct the 3D pose $\boldsymbol{X} \in \mathbb{R}^{3N_{jt} \times T}$ of the center frame, where T is the temporal window size, and N_{jt} is the number of joints in the human model.

When occlusions occur, the 2D joints estimated by human pose detectors are either missing or with low confidence. Such noisy 2D joints adversely affect the accuracy of the 3D pose estimation. Intuitively, we can use attention or soft gated convolution to focus on non-occluded joints in the regression model, which is a well-known technique in image inpainting networks, such as [53, 69, 71]. Similarly, we can also treat 3D pose estimation as a special case of inpainting task when occlusions occur. For the initial input, in addition to feed in the sequential 2D poses, we also feed the binary occlusion mask, $\boldsymbol{M} \in \mathbb{R}^{2N_{jt} \times T}$. Two separate convolution kernels are adopted to obtain the feature maps and soft masks individually. The soft masks can be regarded as soft gating operation or attention on the output features. Specifically, the soft gated convolution is defined as follows,

$$\boldsymbol{Y}^l = \phi\left(\boldsymbol{X}^{l-1} * \boldsymbol{W}_f^l\right),$$

$$\boldsymbol{M}^l = \sigma\left(\boldsymbol{X}^{l-1} * \boldsymbol{W}_g^l\right),$$

$$\boldsymbol{X}^l = \boldsymbol{Y}^l \otimes \boldsymbol{M}^l,$$

(6.1)

where $\boldsymbol{X}^{l-1}$ is the output of the previous layer, $\boldsymbol{W}_f^l$ are the convolution kernels of the feature map, $\boldsymbol{W}_g^l$ are the convolution kernels of the soft mask, $\boldsymbol{Y}^l$ is the feature map of the current layer, $\boldsymbol{M}^l$ is the mask of the soft gate, $\boldsymbol{X}^l$ is the output of the current layer, and ϕ and σ are the activation functions related to feature map and soft mask, respectively. The gated convolution is shown in Fig. 6.1(b).

However, in our experiments, the soft gated convolution defined in Eq. (6.1) does not perform very well. This is reasonable since the soft mask and feature maps are calculated from the same input. It is difficult for the networks to learn different roles of soft gating and feature map extraction at the same time without further supervision. Since the soft gating is more correlated to the input masks rather than the input feature maps, we separate the soft gating and feature map extraction into two streams as shown in Fig. 6.1(c). Specifically, we modify the Eq. (6.1) as follows,

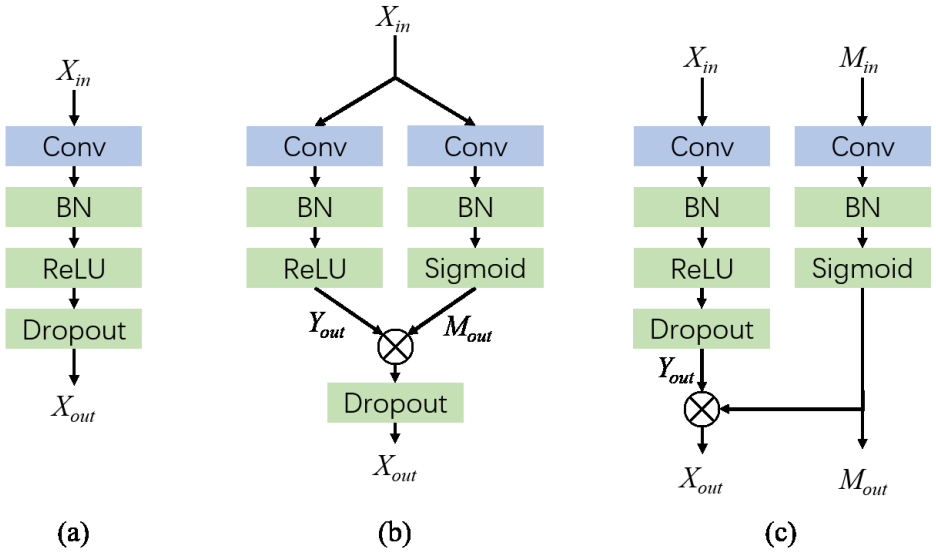


Fig. 6.1. Temporal convolution layer in the pose estimation model. (a): The plain convolution layer used in [63]. (b): The soft gated convolution proposed in [69] for inpainting task. (c): Modified two-stream soft gated convolution proposed in this paper.

$$\begin{aligned}\boldsymbol{Y}^l &= \phi\left(\boldsymbol{X}^{l-1} * \boldsymbol{W}_f^l\right),\\ \boldsymbol{M}^l &= \sigma\left(\boldsymbol{M}^{l-1} * \boldsymbol{W}_g^l\right),\\ \boldsymbol{X}^l &= \boldsymbol{Y}^l \otimes \boldsymbol{M}^l,\end{aligned}$$

(6.2)

where the soft mask is calculated only from the soft mask of the previous layer. This modification can make the network training benefit from easy convergence.

6.1.2 *Network Architecture*

We follow the same regression architecture used in VideoPose3D [63]. Apart from a sequence of 2D joints, we also input the occlusion mask. After the first temporal gated convolution layer, 4 skip blocks are used for regression to estimate the 3D joints. For each layer, we replace the plain convolution as shown in Fig. 6.1(a) with the modified temporal gated convolution as shown in Fig. 6.1(c). For each layer, we input both the feature maps and the soft mask from the previous layer. The temporal dilated convolution is conducted separately for input feature maps and soft mask with different kernels, followed by the batch normalization and activation. We use ReLU for feature map extraction and Sigmoid function for the soft gating. For the supervision, we use the mean square error between predicted 3D joints and ground truth as the loss, which is defined as,

$$\mathcal{L} = \frac{1}{N_{jt}} \sum_{i=1}^{N} \left\| \widehat{\boldsymbol{X}}_i - \boldsymbol{X}_i^* \right\|^2, \tag{6.3}$$

where $\widehat{\boldsymbol{X}}_i$ is the predicted 3D joints and $\boldsymbol{X}_i^*$ is the 3D ground truth.

6.1.3 *Global Pose Trajectory Estimation via Back-Projection*

We adopt the same output format as VideoPose3D [63], i.e. the output of the temporal regression network is the position of all joints with respect to the root location. In VideoPose3D [63], to obtain the global position of each joint, a second network called trajectory model, is trained separately to estimate the trajectory of the human in the camera coordinate. However, this extra effort can be efficiently replaced by a simple optimization approach that utilizes the temporal consistency and 2D-3D back-projection relationship. For simplicity, we will illustrate how to reconstruct the trajectory of one person in the following subsection. For multi-person scenarios, we can estimate all the individual trajectory separately. Denote the root position in the camera coordinate at frame t as $\boldsymbol{C}_t$. Then we can define the projection error as follows,

$$\epsilon_t = \sum_{i=1}^{N_{jt}} \left\| \rho\left(\boldsymbol{X}_{i,t} + \boldsymbol{C}_t\right) - \boldsymbol{x}_{i,t} \right\|^2, \tag{6.4}$$

where ρ is the projection of the pinhole camera, which maps the 3D coordinate into image coordinate, i.e.,

$$\begin{aligned} \rho_x(\boldsymbol{X}) &= \frac{f}{Z} X + c_x, \\ \rho_y(\boldsymbol{X}) &= \frac{f}{Z} Y + c_y. \end{aligned} \tag{6.5}$$

where $\boldsymbol{X} = [X, Y, Z]^T$ represents the 3D point in the camera coordinate, f is the focal length, $\{c_x, c_y\}$ is the center of the image coordinate.

Notice that all the joints share the same global trajectory $\boldsymbol{C}_t$ at each frame. Then the root trajectory $\boldsymbol{C}_t$ can be estimated by minimizing the projection error, i.e.,

$$\widehat{\boldsymbol{C}}_t = arg\min_{\boldsymbol{C}_t} \epsilon_t, \tag{6.6}$$

Since the back-projection is only available for non-occluded joints, the occlusion mask is introduced to constrain the error only on non-occluded joints, i.e.,

$$\epsilon_t = \sum_{i=1}^{N_{jt}} (1 - \boldsymbol{M}_{i,t}) \left\| \rho\left(\boldsymbol{X}_{i,t} + \boldsymbol{C}_t\right) - \boldsymbol{x}_{i,t} \right\|^2, \tag{6.7}$$

where $\boldsymbol{M}_{i,t}$ is defined as

$$\boldsymbol{M}_{i,t} = \begin{cases} 1, & if\ i\ is\ occluded \\ 0, & if\ i\ is\ visible. \end{cases} \tag{6.8}$$

However, if the human is fully occluded in certain frames t, then the projection error ϵ_t becomes zero and solving $\boldsymbol{C}_t$ is impossible. As a result, we adopt the temporal continuity of the person movement and estimate multi-frame trajectory simultaneously as follows,

$$\widehat{\boldsymbol{C}}_t = arg\min_{\boldsymbol{C}_t} \sum_{t=1}^{F} \epsilon_t + \lambda_1 \sum_{t=1}^{F} \|\boldsymbol{C}_t - \boldsymbol{C}_{t-1}\|^2 + \lambda_2 \sum_{t=1}^{F} \|\boldsymbol{C}_{t-1} + \boldsymbol{C}_{t+1} - 2\boldsymbol{C}_t\|^2, \tag{6.9}$$

where λ_1 and λ_2 control the first order and second order temporal continuity, respectively, and F is the number of frames used for estimation. From the Eq. (6.9), the global pose trajectory can be estimated even if full-body occlusions occur.

6.2. Experimental Setup and Results

6.2.1 *Experimental Setup*

a) Datasets

We evaluate the performance of our proposed method on two motion capture datasets, Human3.6M and our recorded dataset MMHuman50K. Following previous work [72, 73, 62, 63, 74, 75, 41, 52] on Human3.6M, we adopt a 17-joint skeleton, training on five subjects (S1, S5, S6, S7, S8), and testing on two subjects (S9 and S11). In our experiments, we consider the following evaluation protocols: Protocol 1 is the mean per-joint position error after alignment with the ground truth in translation, rotation, and scale [63, 62, 74, 75, 52, 65].

A summary of MMHuman50K, which is a subset of our new MMHuman dataset, is listed in Table 6.1.

TABLE 6.1
SUMMARY OF MMHUMAN50K DATASET

Dataset	ShakeHand	WalkCross	HighFive	PullUp	HandOver	Kungfu
Num. Videos	6	6	5	3	5	2
Num. Frames	11,573	11,379	9,537	5,809	9,570	3,883
Total Frames	51,751					

b) Implementation Details for 2D Pose Estimation

To fairly evaluate our performance, all the comparison methods use the same 2D inputs. For experiments on Human3.6M, following VideoPose3D [63] we use fine-tuned cascaded pyramid network (CPN) keypoints [68]. For experiments on MMHuman50K, we use the pre-trained OpenPose [1] to detect 2D keypoints of every frame.

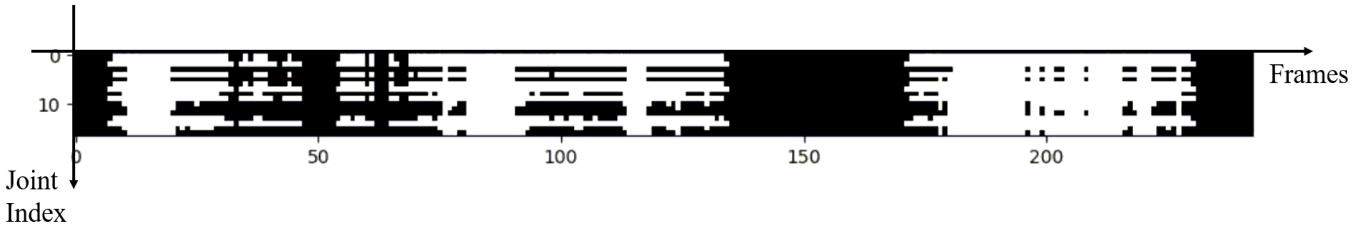


Fig. 6.2. An example of the generated mask in 2D pose estimation. Horizontal and vertical axes represent frame index and joint index, respectively. White color represents the occluded joints while black color represents non-occluded joints.

c) Implementation Details for 3D Pose Estimation

For Human3.6M dataset, we randomly generate occlusion masks in both training and testing on the sequential 2D joints to test the effectiveness of temporal gated convolution on handling long-time occlusion. At first, we generate a matrix P with the same size of the 2D input x with elements sampled from the uniform distribution $U(0,1)$. Then the occlusion mask $\boldsymbol{M}$ can be set as $\boldsymbol{M} = I_{\{\mathrm{P}>\theta\}}$, where I is the indicator function that binarizes P according to the threshold θ. However, it is found that such setting cannot well represent long-time occlusion. The occlusion time span is usually short, and the 2D missing joints can be recovered well enough just by linear interpolation. Instead of using uniformly sampled matrix for binarization directly, we convolve the matrix P with the normalized all-one kernel 1 for each joint dimension separately to mimic long-time occlusion, i.e.,

$$\boldsymbol{M} = \mathbb{I}_{\{\hat{P}>\theta\}}, \tag{6.10}$$

where

$$\hat{P} = \frac{1}{K} P * 1, \tag{6.11}$$

and $*$ represents convolution operation and K is the temporal kernel size. The larger the kernel size K is, the stronger relationship across temporal domain there is. As a result, we can change K to represent the intensity of long-time occlusion. An example of the generated mask is shown in Fig. 6.2. The generated mask can have both partial occlusion and full occlusion.

For evaluation on MMHuman dataset, since occlusions naturally occur in the video recording, we use the originally detected 2D keypoints without generating occlusion masks using the strategy mentioned above. If the confidence of the detected 2D keypoint is below a threshold (we set 0.3 in our experiments), then the joint is treated as under occlusion. Different from Human3.6M dataset, our MMHuman contains multi-person in the video, therefore we apply a multi-object tracking as a pre-processing step for all the comparison methods to solve the association problem. We adopt the TrackletNet tracking (TNT) [31] for human tracking since it is proven very effective when handling complicated multi-object tracking cases with various occlusions for both static and moving cameras' recording. The details of TNT can be found in [31].

For the comparison approaches listed in the following section, missing joints from the inputs could cause trouble in the inference. Therefore, we use linear interpolation in the temporal domain to fill out the missing inputs instead of directly feeding incomplete data. This gives us the best performances we could possibly get using the comparison approaches.

d) Hyper Parameter Setting

For the temporal gated convolution model, we use Amsgrad [76] as the optimizer and train for 80 epochs. We set the batch size as 1024 in the training. For Human3.6M, we adopt an exponentially decaying learning rate schedule, starting from $\eta = 0.001$ with a shrink factor $\alpha = 0.95$ applied on each epoch. For the global pose trajectory estimation, we set $\lambda_1 = 0.01, \lambda_2 = 0.01$ and use $F = 100$ frames in the optimization.

6.2.2 *Quantitative Results for Occlusion Handling*

We compare our method with recent state-of-the-art temporal based models [65, 66, 63], where both open source codes with pre-trained models provided. To explore severe occlusion cases, we randomly generate binary masks with occlusion ratio equaling to 50% in our experiments, i.e., 50% input joints are treated as occluded joints. Table 6.2 shows the 3D pose error following protocol 1 on Human3.6M dataset. From the table, we can see our proposed method achieves promising results. It shows the effectiveness of using gated convolution to handle heavy occlusion in the 3D pose estimation.

TABLE 6.2
3D POSE ERROR, OCCLUSION RATIO EQUALING TO 50% (PROTOCOL 1, ERRORS IN MM).

Method	Class							
	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchase
Hossain et al.[65]	50.8	60.3	59.6	60.1	75.0	67.9	53.7	59.8
Pavllo et al.[63]	46.3	53.9	49.3	56.8	48.2	57.7	51.2	51.5
Lin et al.[66]	51.3	52.5	49.7	55	47.1	64.9	47.8	50.2
Proposed	**42.9**	**43.5**	**41.5**	**44.7**	**41.9**	**49.4**	**42**	**41**
	Sitting	Sitting Down	Smoke	Wait	Walk Dog	Walk	Walk Together	Avg.
Hossain et al.[65]	71.1	84.7	62.5	59.2	68.1	69.9	72	65
Pavllo et al.[63]	53	63.6	50.8	45.5	56.3	57	56.2	53.2
Lin et al.[66]	53.1	61.7	51	47.6	60.1	65.7	62.1	53.9
Proposed	**50.4**	**59.8**	**43.9**	**41.3**	**46.5**	**35.1**	**32.6**	**43.8**

To further analyze the impact of occlusions in the input 2D pose, We also report the pose reconstruction errors on Human3.6M under protocol 1 with different occlusion ratios in the generated mask in Table 6.3. The results meet our expectation. As the occlusion ratio increases, all methods have higher errors in the 3D pose estimation. When there are 50% percent of missing joints, our method achieves a significantly better performance, reporting an average error of 43.8 mm, which is 9.4 mm better than the second best. Moreover, our proposed method does not have a large fluctuation on the reconstruction errors given different setting of occlusion ratios. It shows the robustness of the proposed method facing different occlusion situations. Note that for the first column in the table when no occlusion occurs, the reconstruction error of the proposed method is slightly higher (about 0.7 mm) than VideoPose3D [63]. This is because since we add a large amount of challenging examples with occlusions during training, the data distribution is slightly different between training set and testing set if no occlusion occurs in the testing time. This will lead to a slightly higher error than VideoPose3D [63]. Besides that, when no occlusion occurs, the input mask becomes a zero-matrix, and the advantage of gated convolution is not well utilized.

TABLE 6.3
3D POSE ERROR ON HUMAN3.6M WITH DIFFERENT OCCLUSION RATIOS (ERRORS IN MM).

Occlusion Ratio	0%	25%	50%
Hossain et al. [65]	44.1	57.2	65.0
Pavllo et al.[63]	36.5	40.3	53.2
Lin et al. [66]	36.8	41.9	53.9
Proposed	**37.2**	**39.2**	**43.8**

Table 6.4 shows the results on our recorded dataset MMHuman50K. This dataset, as explained in Chapter 4, includes more inter-person occlusions during human interactions. For this dataset, each detected 2D keypoint by OpenPose is associated with a confidence score. We treat it as occlusion if the confidence below 0.3. To test the generalization of the methods, we use the pre-trained model on Human3.6M and not fine-tuned on MMHuman dataset. From Table 6.4, we can see that our proposed method achieves the best

performance. It further proves that our pose estimation method is more robust facing diverse real-world scenarios.

TABLE 6.4
3D POSE ERROR ON MMHUMAN (ERRORS IN MM)

	ShakeHand	Walk Cross	HighFive	PullUp	HandOver	Kungfu	Avg.
Hossain et al. [65]	119.3	119.1	119.1	120.2	115.2	119.3	118.5
Pavllo et al. [63]	87.7	86.1	82.8	88.8	79.1	78.7	84.2
Lin et al. [66]	110.5	116.3	123.2	116.3	107.7	118.4	114.9
Proposed	83.7	74.8	78.6	83.9	75.1	76.7	78.3

6.2.3 *Qualitative Results for Occlusion Handling*

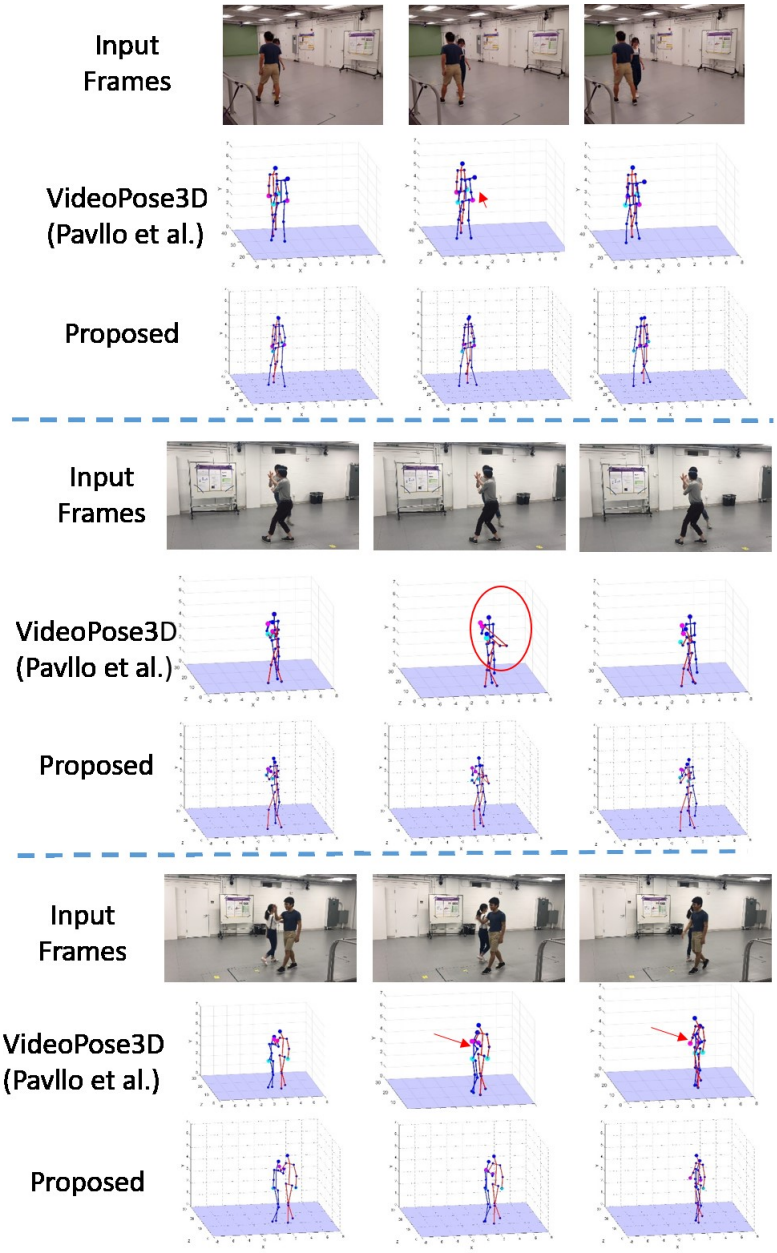


Fig. 6.3. Qualitative results on MMHuman50K.

Fig. 6.3 shows some qualitative results of our proposed method. The recovered 3D poses using our proposed method as well as state-of-the-art method VideoPose3D are displayed in the camera coordinate. Each example has either severe partial occlusion or full-body occlusion, which is a common problem in human pose estimation the proposed method is targeted at. The first sequence (belonging to ”PullUp”) shows that being at the end of an inter-person occlusion period, our proposed method demonstrates reliable performance while VideoPose3D gets adversely affected. The second sequence (”Kungfu”) shows under severe inter-person occlusion, the proposed method still outputs reasonable result, whereas VideoPose3D messes up. For the third (”HighFive”) sequence, VideoPose3D gives unnatural limb estimations, whereas the proposed method predicts natural poses that are consistent with temporal context. It demonstrates that the proposed method is clearly more reliable than other competing methods in occlusion cases.

6.3. Discussion

To recover natural motion, we proposed a temporal gated convolution model for 3D human pose estimation to address severe occlusion issues in the real-world scenario in this chapter. Meanwhile, the global pose trajectory is efficiently and effectively estimated via temporal back-projection between 2D and 3D joint sequences. We tested our proposed model on scenarios with occlusions, and explore the performance on different severity levels of occlusions. We outperform several state-of-the-art 3D pose estimation methods with temporal models on Human3.6M under occlusions, and also our newly proposed dataset MMHuman.

7. CONCLUSIONS AND FUTURE WORK

In this dissertation, we address the problem of multi-person human pose estimation in natural videos. We propose an efficient unsupervised method, Hierarchical 3D Human Pose Estimation (H3DHPE). Our framework first utilizes a 2D keypoint detector, associates and tracks multiple people, and then estimating each person's 3D human poses hierarchically with body geometric constraints. When estimating each person's poses, we apply a prior flexible human model that contains bone lengths of all human body parts, and allows bone lengths to be optimized. Instead of trying to solve all the poses in high dimensions simultaneously, we first estimate the torso pose, and then estimate limb poses in a hierarchical fashion. We extend this work to Universal Hierarchical 3D Human Pose Estimation (UH3DHPE) and propose a direct bottom-up pose estimation alternative under the case of unreliable torso keypoints. A new natural human interaction dataset, MMHuman, is collected for our evaluation, and for the benefit of future research. To recover trajectory for severe and/or long-term occlusion, we also explore a DNN based solution that uses temporal gated convolutions to recover missing poses and address the occlusion issues in the pose estimation.

For future work, there are several promising directions. For unsupervised solutions, along the line of H3DHPE and UH3DHPE, 1) Design the adaptive weights merging strategy (AWMS) according to keypoints confidence, and weigh each term inversely proportional to the objective function, instead of using binary flags of optimization failure. 2) When estimating human torso pose, model the torso more generically, instead of modeling it as a trapezoid, so that poses like twisting body can be better modeled.

For supervised solutions that uses training data, 1) we can consider combining the recent adversarial learning models to enhance the robustness of the 3D pose reconstruction with occlusion. 2) Apply angle constraints and/or bone length constraints during training.

Finally, combining supervised solutions with unsupervised solutions, we may initialize 3D pose estimates with output of a trained model, and exploit optimization to achieve optimum solution in the following step.

8. REFERENCES

[1] Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, Raquel Urtasun, Vision meets Robotics: The KITTI Dataset, International Journal of Robotics Research (IJRR), 2013.

[3] Ess, A., Leibe, B., Schindler, K. and Van Gool, L. Robust multiperson tracking from a mobile platform. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10), pp.1831-1846, 2009.

[4] Weinzaepfel, Philippe, Xavier

Martin, and Cordelia Schmid. Human Action Localization with Sparse Spatial Supervision. arXiv preprint arXiv:1605.05197, 2016.

[5] Catalin Ionescu, Dragos Papava, Vlad Olaru and Cristian Sminchisescu, Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, No. 7, July 2014.

[6] Wei, S.E., Ramakrishna, V., Kanade, T. and Sheikh, Y. Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4724-4732), 2016.

[7] Chen, X. and Yuille, A.L. Articulated pose estimation by a graphical model with image dependent pairwise relations. In Advances in neural information processing systems (pp. 1736-1744), 2014.

[8] Ouyang, W., Chu, X. and Wang, X. Multi-source deep learning for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2329-2336), 2014.

[9] Tompson, J.J., Jain, A., LeCun, Y. and Bregler, C. Joint training of a convolutional network and a graphical model for human pose estimation. In Advances in neural information processing systems (pp. 1799-1807), 2014.

[10] Toshev, A. and Szegedy, C. Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1653-1660), 2014.

[11] Fan, X., Zheng, K., Lin, Y. and Wang, S. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1347-1355), 2015.

[12] Bulat, A. and Tzimiropoulos, G. Human pose estimation via convolutional part heatmap regression. In European Conference on Computer Vision (pp. 717-732). Springer, Cham, 2016.

[13] Newell, A., Yang, K. and Deng, J. Stacked hourglass networks for human pose estimation. In European Conference on Computer Vision (pp. 483-499). Springer, Cham, 2016.

[14] Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L. and Wang, X. Multi-context attention for human pose estimation. arXiv preprint arXiv:1702.07432, 1(2) , 2017.

[15] Chen, Y., Shen, C., Wei, X.S., Liu, L. and Yang, J. Adversarial posenet: A structure-aware convolutional network for human pose estimation. CoRR, abs/1705.00389, 2, 2017.

[16] Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B., 2d human pose estimation: New benchmark and state of the art analysis, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

[17] Sidenbladh, H., Black, M.J. and Fleet, D.J., Stochastic tracking of 3D human figures using 2D image motion. In European conference on computer vision (pp. 702-718). Springer, Berlin, Heidelberg, 2000.

[18] Li, S. and Chan, A.B. 3d human pose estimation from monocular images with deep convolutional neural network. In Asian Conference on Computer Vision (pp. 332-347). Springer, Cham, 2014.

[19] Li, S., Zhang, W. and Chan, A.B. Maximum-margin structured learning with deep networks for 3d human pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2848-2856), 2015.

[20] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J. and Black, M.J. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In European Conference on Computer Vision (pp. 561-578). Springer, Cham, 2016.

[21] Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C. and Moreno-Noguer, F., 2012, June. Single image 3D human pose estimation from noisy observations. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (pp. 2673-2680), 2012.

[22] Ramakrishna, V., Kanade, T. and Sheikh, Y. Reconstructing 3d human pose from 2d image landmarks. In European conference on computer vision (pp. 573-586). Springer, Berlin, Heidelberg, 2012.

[23] Wang, C., Wang, Y., Lin, Z., Yuille, A.L. and Gao, W. Robust estimation of 3d human poses from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2361-2368), 2014.

[24] Wang, C., Wang, Y., Lin, Z. and Yuille, A Robust 3D Human Pose Estimation from Single Images or Video Sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.

[25] Hashim Yasin, Umar Iqbal, Björn Krüger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D pose estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4948-4956), 2016.

[26] Simo-Serra, E., Quattoni, A., Torras, C. and Moreno-Noguer, F. A joint model for 2d and 3d pose estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3634-3641), 2013.

[27] Zhou, F. and De la Torre, F, Spatio-temporal matching for human detection in video. In European Conference on Computer Vision (pp. 62-77). Springer, Cham, 2014.

[28] Gregory Rogez, Philippe Weinzaepfel, Cordelia Schmid, LCR-Net: Localization-Classification-Regression for Human Pose, CVPR, 2017.

[29] Bastian Wandt, Hanno Ackermann, and Bodo Rosenhahn, 3D Reconstruction of Human Motion from Monocular Image Sequences, TPAMI, AUGUST 2016.

[30] Zhou, Xingyi and Huang, Qixing and Sun, Xiao and Xue, Xiangyang and Wei, Yichen, Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach, The IEEE International Conference on Computer Vision (ICCV), 2017.

[31] Wang, G., Wang, Y., Zhang, H., Gu, R. and Hwang, J.N. Exploit the connectivity: Multi-object tracking with trackletnet. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 482-490), 2019.

[32] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, Davide Scaramuzza, SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems, IEEE transaction on Robotics, 2017.

[33] Lee, K.H. and Hwang, J.N. On-road pedestrian tracking across multiple driving recorders. IEEE Transactions on Multimedia, 17(9), pp.1429-1438, 2015.

[34] Gao, X.S., Hou, X.R., Tang, J. and Cheng, H.F. Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence, 25(8), pp.930-943, 2003.

[35] Powell, M.J. An efficient method for finding the minimum of a function of several variables without calculating derivatives. The computer journal, 7(2), pp.155-162, 1964.

[36] Akhter, I. and Black, M.J. Pose-conditioned joint angle limits for 3D human pose reconstruction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1446-1455), 2015.

[37] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K., Daniilidis, K.: Sparse representation for 3D shape estimation: A convex relaxation approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4447–4455, 2015.

[38] Hamilton, N.P., 2011. Kinesiology: scientific basis of human motion. Brown & Benchmark.

[39] Pavlakos, G., Zhou, X., Derpanis, K.G. and Daniilidis, K.. Coarse-to-fine volumetric prediction for single-image 3D human pose. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on (pp. 1263-1272), 2017.

[40] Rogez, G., Weinzaepfel, P. and Schmid, C. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. IEEE transactions on pattern analysis and machine intelligence, 2019.

[41] Luvizon, D.C., Picard, D. and Tabia, H., 2d/3d pose estimation and action recognition using multitask deep learning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018 June.

[42] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," IJCV, 2010.

[43] CMU motion capture dataset.http://mocap.cs.cmu.edu.

[44] W. Chen, H. Wang, Y.Li, H.Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen, "Synthesizing training images for boosting human 3D pose estimation," in 3DV, 2016.

[45] G. Rogez and C. Schmid, "MoCap-guided data augmentation for 3D pose estimation in the wild," in NIPS, 2016.

[46] Tianmin Shu, M. S. Ryoo and Song-Chun Zhu. Learning Social Affordance for Human-Robot Interaction. International Joint Conference on Artificial Intelligence (IJCAI), 2016.

[47] Xiaohan Nie, B., Wei, P. and Zhu, S.C., Monocular 3D human pose estimation by predicting depth on joints. In Proceedings of the IEEE International Conference on Computer Vision, 2017.

[48] Qualysis: https://www.qualisys.com/

[49] Ke Sun, Bin Xiao, Dong Liu, Jingdong Wang, Deep High-Resolution Representation Learning for Human Pose Estimation, CVPR 2019.

[50] Iqbal, U. and Gall, J., Multi-person pose estimation with local joint-to-person associations. In European Conference on Computer Vision (pp. 627-642) 2016.

[51] Renshu Gu, Gaoang Wang, Jenq-neng Hwang, Efficient Multi-Person Hierarchical 3D Pose Estimation for autonomous driving, MIPR 2019.

[52] Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H. and Wang, X, 3d human pose estimation in the wild by adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018.

[53] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. and Huang, T.S., Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018.

[54] Kevin J. McQuade, Renshu Gu, Jenq-neng Hwang, Using kinect sensors for determining Functional Reach Volume as an upper-extremity functional outcome measure, XXVI Congress of the International Society of Biomechanics, 2017.

[55] Zheng Tang, Renshu Gu, Jenq-Neng Hwang, Joint Multi-View People Tracking and Pose Estimation for 3D Scene Reconstruction, ICME 2018.

[56] Sigal, L., Black, M.J.: Humaneva: Synchronized video and motion capture datasetfor evaluation of articulated human motion. Brown Univertsity TR 120 (2006)

[57] Trumble, M., Gilbert, A., Malleson, C., Hilton, A. and Collomosse, J. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In BMVC (Vol. 2, p. 3), 2017.

[58] Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C. Monocular 3d human pose estimation in the wild using improved cnn supervision. International Conference on 3D Vision (3DV). pp. 506–516. IEEE (2017)

[59] von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G. Recovering accurate 3d human pose in the wild using imus and a moving camera. Proceedings of the European Conference on Computer Vision (ECCV). pp. 601–617 (2018)

[60] Kanazawa, A., Black, M.J., Jacobs, D.W. and Malik, J. End-to-end recovery of human shape and pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7122-7131), 2018.

[61] Zhou, X., Huang, Q., Sun, X., Xue, X. and Wei, Y. Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach, The IEEE International Conference on Computer Vision (ICCV), 2017.

[62] Martinez, J., Hossain, R., Romero, J. and Little, J.J., 2017. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2640-2649).

[63] Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7753–7762, 2019.

[64] Wang, G., Wang, Y., Zhang, H., Gu, R. and Hwang, J.N., Exploit the connectivity: Multi-object tracking with trackletnet. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 482-490), 2019.

[65] Rayat Imtiaz Hossain, M. and Little, J.J., 2018. Exploiting temporal information for 3d human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 68-84).

[66] Lin, J. and Lee, G.H., 2019. Trajectory Space Factorization for Deep Video-Based 3D Human Pose Estimation. arXiv preprint arXiv:1908.08289.

[67] Kanazawa, A., Zhang, J.Y., Felsen, P. and Malik, J., 2019. Learning 3d human dynamics from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5614-5623).

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

[69] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4471–4480 (2019).

[70] Gu, R., Wang, G., Jiang, Z. and Hwang, J.N. Multi-Person Hierarchical 3D Pose Estimation in Natural Videos. IEEE Transactions on Circuits and Systems for Video Technology, 2019.

[71] Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[72] Pavlakos, G., Zhou, X., Derpanis, K.G. and Daniilidis, K. Coarse-to-fine volumetric prediction for single-image 3D human pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7025-7034), 2017.

[73] Tekin, B., Márquez-Neila, P., Salzmann, M. and Fua, P., 2017. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 3941-3950).

[74] Sun, X., Shang, J., Liang, S. and Wei, Y., 2017. Compositional human pose regression. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2602-2611).

[75] Fang, H.S., Xu, Y., Wang, W., Liu, X. and Zhu, S.C. Learning pose grammar to encode human body configuration for 3d pose estimation. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[76] Reddi, S.J., Kale, S., Kumar, S.: On the convergence of adam and beyond. arXivpreprint arXiv:1904.09237, 2019.

[77] Arnab, A., Doersch, C. and Zisserman, A. Exploiting temporal context for 3D human pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3395-3404), 2019.

[78] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P. and Suleyman, M. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.