Ph.D. DISSERTATION

Expressive Whole-Body 3D Multi-Person Pose and Shape Estimation from a Single Image

단일 이미지로부터 여러 사람의 표현적 전신 3D 자세 및 형태 추정

BY

GYEONGSIK MOON

February 2021

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Expressive Whole-Body 3D Multi-Person Pose and Shape Estimation from a Single Image

단일 이미지로부터 여러 사람의
표현적 전신 3D 자세 및 형태 추정

지도교수 이 경 무
이 논문을 공학박사 학위논문으로 제출함
2021년 2월

서울대학교 대학원
전기정보공학부
문 경 식

문경식의 공학박사 학위논문을 인준함
2021년 2월

위 원 장 한 보 형 (인)
부 위 원 장 이 경 무 (인)
위 원 전 건 희 (인)
위 원 김 희 호 (인)
위 원 윤 국 진 (인)

Abstract

Human is the most centric and interesting object in our life: many human-centric techniques and studies have been proposed from both industry and academia, such as motion capture and human-computer interaction. Recovery of accurate 3D geometry of human (*i.e.*, 3D human pose and shape) is a key component of the human-centric techniques and studies. With the rapid spread of cameras, a single RGB image has become a popular input, and many single RGB-based 3D human pose and shape estimation methods have been proposed.

The 3D pose and shape of the whole body, which includes hands and face, provides expressive and rich information, including human intention and feeling. Unfortunately, recovering the whole-body 3D pose and shape is greatly challenging; thus, it has been attempted by few works, called expressive methods. Instead of directly solving the expressive 3D pose and shape estimation, the literature has been developed for recovery of the 3D pose and shape of each part (*i.e.*, body, hands, and face) separately, called part-specific methods. There are several more simplifications. For example, many works estimate only 3D pose without shape because additional 3D shape estimation makes the problem much harder. In addition, most works assume a single person case and do not consider a multi-person case. Therefore, there are several ways to categorize current literature; 1) part-specific methods and expressive methods, 2) 3D human pose estimation methods and 3D human pose and shape

estimation methods, and 3) methods for a single person and methods for multiple persons. The difficulty increases while the outputs of methods become richer by changing from part-specific to expressive, from 3D pose estimation to 3D pose and shape estimation, and from a single person case to multi-person case.

This dissertation introduces three approaches towards expressive 3D multi-person pose and shape estimation from a single image; thus, the output can finally provide the richest information. The first approach is for 3D multi-person body pose estimation, the second one is 3D multi-person body pose and shape estimation, and the final one is expressive 3D multi-person pose and shape estimation. Each approach tackles critical limitations of previous state-of-the-art methods, thus bringing the literature closer to the real-world environment.

First, a 3D multi-person body pose estimation framework is introduced. In contrast to the single person case, the multi-person case additionally requires camera-relative 3D positions of the persons. Estimating the camera-relative 3D position from a single image involves high depth ambiguity. The proposed framework utilizes a deep image feature with the camera pinhole model to recover the camera-relative 3D position. The proposed framework can be combined with any 3D single person pose and shape estimation methods for 3D multi-person pose and shape. Therefore, the following two approaches focus on the single person case and can be easily extended to the multi-person case by using the framework of the first approach. Second, a 3D multi-person body pose and shape estimation method is introduced. It extends the first approach to additionally predict accurate 3D shape while its accuracy significantly outperforms previous state-of-the-art methods by proposing a new target representation, lixel-based 1D heatmap. Finally, an expressive 3D multi-person pose and shape estimation method is introduced. It integrates the part-specific 3D pose

and shape of the above approaches; thus, it can provide expressive 3D human pose and shape. In addition, it boosts the accuracy of the estimated 3D pose and shape by proposing a 3D positional pose-guided 3D rotational pose prediction system.

The proposed approaches successfully overcome the limitations of the previous state-of-the-art methods. The extensive experimental results demonstrate the superiority of the proposed approaches in both qualitative and quantitative ways.

Key words: 3D human pose, 3D human shape, expressive whole-body, multiple persons, single image

Student number: 2015-22785

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Research Issues

From the distant past, human has been the most centric and interesting object in our life. Mythological gods are depicted in a human form, and ancient murals and medieval art depict people to show the situation at that time. In modern society, many interesting human-centric techniques, such as motion capture, virtual try-on, and AR/VR, have been introduced, thanks to highly advanced computer graphics, computer vision, and human-computer interaction, as shown in Figure 1.1. Recovering accurate 3D geometry of human (*i.e.*, 3D human pose and shape) is a key component of the recent techniques; thus, 3D human pose and shape estimation has drawn significant attention from both industry and academia. In particular, the rapid popularization of cameras and smartphones motivated many single RGB-based 3D human pose and shape estimation methods. Their goal is to recover the 3D pose and shape of humans in the input image, mostly taken from a single camera in an uncontrolled environment (*i.e.*, in-the-wild images).

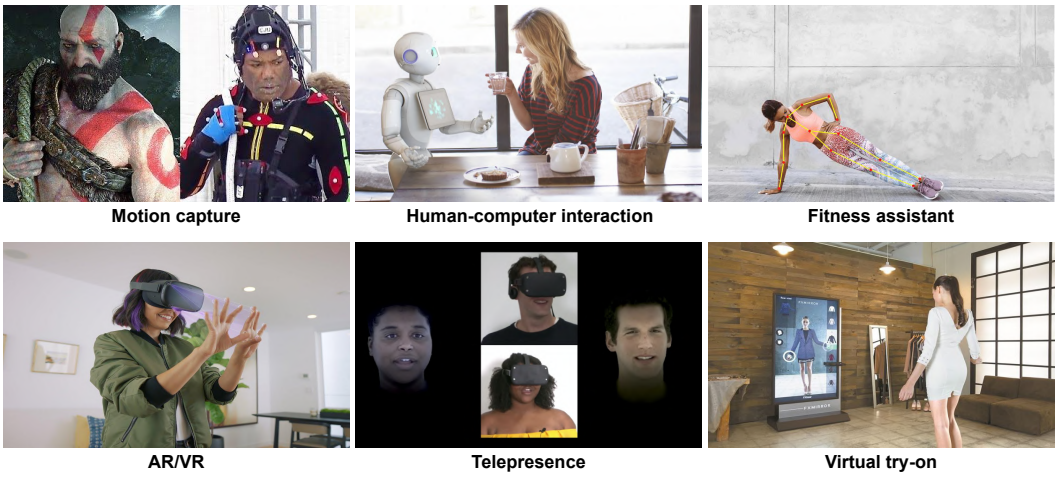


Figure 1.1: 3D human pose and shape estimation is a key component of modern human-centric techniques. From left to right and top to bottom, the figures are from 1, 2, 3, 4, 5, and 6.

We, humans, deliver our intention and feeling through a combination of body and hand motions with a facial expression. Therefore, the 3D pose and shape of the whole body, which includes hands and face, provides greatly rich information. Unfortunately, recovering the whole-body 3D pose and shape is highly challenging; thus, it has been attempted by very few works, called expressive methods [6,7,19,20]. Instead of directly solving the expressive 3D pose and shape estimation, the literature has been developed to recover the 3D pose and shape of each part (*i.e.*, body, hands, and face) separately, called part-specific methods [5, 21–25]. Other than this, there are several more simplifications. For example, many works estimate only 3D pose without shape because additional 3D shape estimation makes the problem much more difficult [3, 14, 26–28]. In addition, most previous works assume a single person case [5, 23–25, 28–33] and do not consider a multi-person case. Therefore, there are several ways to categorize current literature; 1) part-specific methods and expressive methods, 2) 3D human pose estimation methods and 3D human pose and shape estimation methods, and 3) methods for a single person and methods for multiple

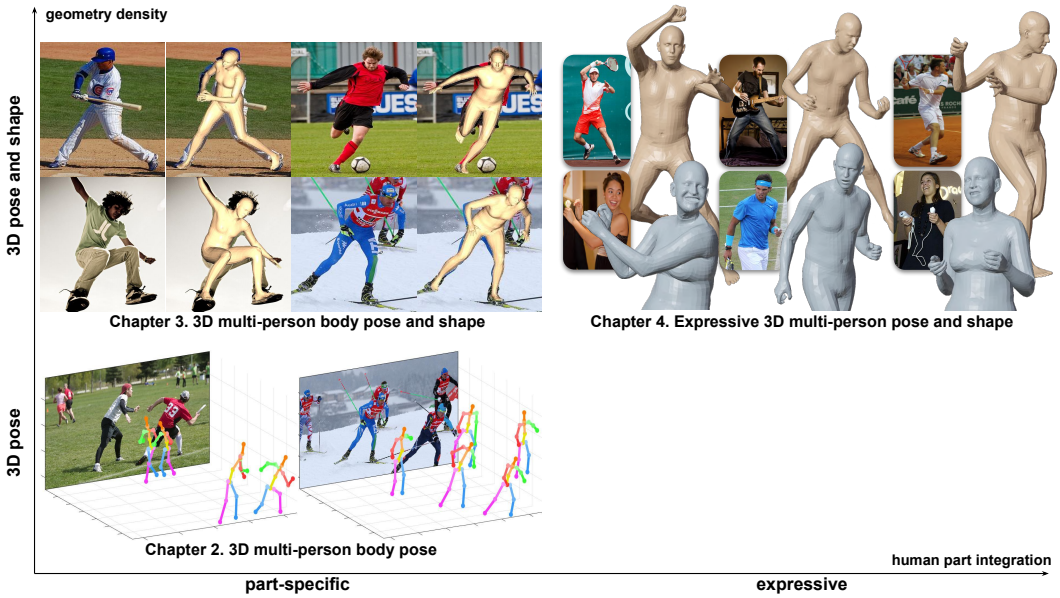


Figure 1.2: The research direction of this dissertation. Starting from the part-specific 3D pose estimation in Chapter 2, part-specific 3D pose and shape estimation is introduced in Chapter 3. Finally, expressive 3D pose and shape estimation is introduced in Chapter 4, which provides the richest information, including human intention and feeling. Note that the 3D pose and shape of a single person in Chapter 3 and Chapter 4 can be easily extended to the multi-person case by combining them with the framework of Chapter 2.

persons. The difficulty increases while the outputs of methods become richer by changing from part-specific to expressive, from 3D pose estimation to 3D pose and shape estimation, and from a single person case to multi-person case.

1.2 Outline of the Dissertation

This dissertation introduces three approaches towards expressive 3D multi-person pose and shape estimation from a single image; thus, the output can finally provide the richest information, as shown in Figure 1.2. The first approach is for 3D multi-person body pose estimation, the second one is 3D multi-person body pose and shape estimation, and the final one is expressive 3D multi-person pose and shape estimation. Each approach tackles critical limitations of previous state-of-the-art

methods, thus bringing the literature closer to the real-world environment.

In Chapter 2, a camera distance-aware 3D multi-person body pose estimation framework [34] is introduced. As described above, extending the 3D single body pose estimation to the multi-person case requires the relative 3D position between the camera and persons, which involves high depth ambiguity. To this end, I propose RootNet, which refines a pre-defined human scale (2 meters $\times$ 2 meters) using a deep image feature. The human scale, pre-defined for grown-ups with a standing pose, can change according to the pose and appearance of a person in the input image. For example, taking a crouching pose makes the scale smaller, and a child has a smaller scale. As the deep image feature contains information on both pose and appearance, RootNet can successfully refine the human scale. The refined human scale is plugged into the camera pinhole model equation, which provides the relative position between the camera and person. By combining RootNet with state-of-the-art human detection and 3D single person body pose estimation methods, 3D multi-person body pose is successfully recovered from in-the-wild images. Another advantage of the proposed framework is that it can be combined with any 3D single person pose and shape estimation methods. Thus, the following two approaches focus on the single person case, and their outputs are easily extended to the multi-person case by combining them with the proposed framework.

In Chapter 3, a 3D human pose and shape estimation method, I2L-MeshNet [21], is introduced. It extends 3D pose estimation methods of the first chapter to additionally predict 3D shape (*i.e.*, 3D mesh) while preserving their accuracy. Most of the previous 3D human pose and shape estimation methods [5, 23–25] directly regress 3D rotations of human joints, which is a highly non-linear mapping as addressed by Moon *et al.* [35]. To resolve this issue, I propose the use of a heatmap

as a prediction target instead of the 3D rotations. The heatmap preserves the spatial relationship between pixels in the input image and can model uncertainty, thus making it easier for the estimator to predict as demonstrated by Moon *et al.* [35]. However, unlike human joints consisting of several points, human mesh consists of more than thousands of vertices. Thus, simply extending the 3D pose estimation methods of predicting a 3D heatmap for each joint [9, 34] to mesh vertices causes drastic GPU memory usage. To resolve this issue, I design I2L-MeshNet to predict three lixel-based 1D heatmaps for each mesh vertex in x-, y-, and z-axis instead of predicting a voxel-based 3D heatmap. The lixel (line+pixel) is a quantized cell in one-dimensional space; likewise, voxel (volume+pixel) is defined as a quantized cell in three-dimensional space. I show that the proposed I2L-MeshNet is much more efficient than networks that predict the voxel-based 3D heatmap while achieving better accuracy under a similar number of learnable parameters.

Finally, in Chapter 4, expressive 3D human pose and shape estimation method, Pose2Pose [20], is introduced. Although the above described I2L-MeshNet [21] achieves high accuracy, 3D rotations of human joints are needed for many computer graphics applications, such as animation. Thus, Pose2Pose is designed to improve the accuracy of 3D rotation of human joints (*i.e.*, 3D rotational pose) prediction by using the heatmap as guidance. Unlike previous works [5, 23–25] that only rely on global image features when predicting the 3D rotational pose, Pose2Pose utilizes joint-specific local and global features, extracted from positions of human joints (*i.e.*, positional pose) when predicting the 3D rotational pose, where the positional pose is from the heatmap. In addition, the proposed framework integrates the 3D poses and shapes of body/hands with a facial expression; thus, it can provide expressive 3D human pose and shape and convey human intention and feeling, while most of the previous

works [5, 21, 23–25] can recover only one of body, hands, and face. The experimental results demonstrate the superiority of Pose2Pose in both qualitative and quantitative ways.

The conclusion of the dissertation is provided in Chapter 5 with a summary, and suggestions for future works are also provided.

Chapter 2

3D Multi-Person Pose Estimation

2.1 Introduction

The goal of 3D multi-person body pose estimation is to localize semantic keypoints of multiple human bodies in 3D space. Recently, many methods [28–33] utilize deep convolutional neural networks (CNNs) and have achieved noticeable performance improvement on large-scale publicly available datasets [2, 14].

Most of the previous 3D human pose estimation methods [28–33] are designed for single-person case. They crop the human area in an input image with a groundtruth bounding box or the bounding box that is predicted from a human detection model [36]. The cropped patch of a human body is fed into the 3D pose estimation module, which then estimates the 3D location of each keypoint. As their models take a single cropped image, estimating the absolute camera-centered coordinate of each keypoint is difficult. To handle this issue, many methods [28–33] estimate the relative 3D pose

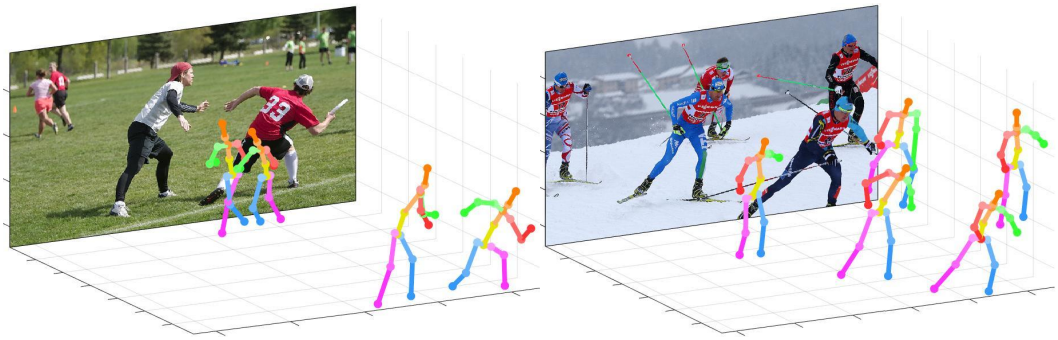


Figure 2.1: Qualitative results of applying our 3D multi-person pose estimation framework to COCO dataset [1] which consists of *in-the-wild* images. Most of the previous 3D human pose estimation studies mainly focused on the root-relative 3D single-person pose estimation. In this study, we propose a general 3D *multi*-person pose estimation framework that takes into account all factors including human detection and 3D human root localization.

to a reference point in the body, e.g., the center joint (*i.e.*, pelvis) of a human, called *root*. The final 3D pose is obtained by adding the 3D coordinates of the root to the estimated root-relative 3D pose. Prior information on the bone length [29] or the groundtruth [28] has been commonly used for the localization of the root.

Recently, many top-down approaches [18, 37, 38] for the 2D multi-person pose estimation have shown noticeable performance improvement. These approaches first detect humans by using a human detection module and then estimate the 2D pose of each human by a 2D single-person pose estimation module. Although they are straightforward when used in 2D cases, extending them to 3D cases is nontrivial. Note that for the estimation of 3D multi-person poses, we need to know the absolute distance to each human from the camera as well as the 2D bounding boxes. However, existing human detectors provide 2D bounding boxes only.

In this study, we propose a general framework for 3D multi-person pose estimation. To the best of our knowledge, this study is the first to propose a fully learning-based camera distance-aware top-down approach of which components are compatible with most of the previous human detection and 3D human pose estima-

tion methods. The pipeline of the proposed system consists of three modules. First, a human detection network (DetectNet) detects the bounding boxes of humans in an input image. Second, the proposed 3D human root localization network (RootNet) estimates the camera-centered coordinates of the detected humans' roots. Third, a root-relative 3D single-person pose estimation network (PoseNet) estimates the root-relative 3D pose for each detected human. Figures 2.1 and 2.2 show the qualitative results and overall pipeline of our framework, respectively.

We show that our approach outperforms previous 3D multi-person pose estimation methods [3, 13] on several publicly available 3D single- and multi-person pose estimation datasets [2, 3] by a large margin. Also, even without any groundtruth information (*i.e.*, the bounding boxes and the 3D location of the roots), our method achieves comparable performance with the state-of-the-art 3D single-person pose estimation methods that use the groundtruth in the inference time. Note that our framework is new but follows previous conventions of object detection and 3D human pose estimation networks. Thus, previous detection and pose estimation methods can be easily plugged into our framework, which makes the proposed framework quite flexible and generalizable.

Our contributions can be summarized as follows.

- We propose a new general framework for 3D multi-person pose estimation from a single RGB image. The framework is the first fully learning-based, camera distance-aware top-down approach, of which components are compatible with most of the previous human detection and 3D human pose estimation models.

- Our framework outputs the absolute camera-centered coordinates of multiple humans' keypoints. For this, we propose a 3D human root localization network

(RootNet). This model makes it easy to extend the 3D single-person pose estimation techniques to the absolute 3D pose estimation of multiple persons.

- We show that our method significantly outperforms previous 3D multi-person pose estimation methods on several publicly available datasets. Also, it achieves comparable performance with the state-of-the-art 3D single-person pose estimation methods without any groundtruth information.

2.2 Related works

2D multi-person pose estimation. There are two main approaches in the multi-person pose estimation. The first one, the top-down approach, deploys a human detector that estimates the bounding boxes of humans. Each detected human area is cropped and fed into the pose estimation network. The second one, the bottom-up approach, localizes all human body keypoints in an input image first and then groups them into each person using some clustering techniques.

[18,37–41] are based on the top-down approach. Papandreou *et al.* [39] predicted 2D offset vectors and 2D heatmaps for each joint. They fused the estimated vectors and heatmaps to generate highly localized heatmaps. Chen *et al.* [38] proposed a cascaded pyramid network whose cascaded structure refines an initially estimated pose by focusing on hard keypoints. Xiao *et al.* [18] used a simple pose estimation network that consists of a deep backbone network and several upsampling layers.

[42–46] are based on the bottom-up approach. Cao *et al.* [44] proposed the part affinity fields (PAFs) that model the association between human body keypoints. They grouped the localized keypoints of all persons in the input image by using the estimated PAFs. Newell *et al.* [45] introduced a pixel-wise tag value to assign

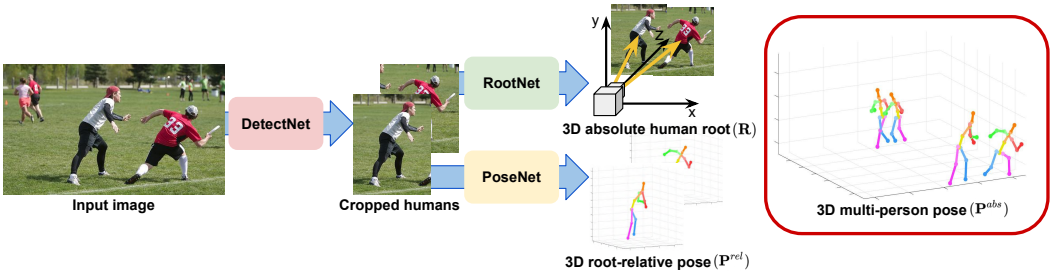


Figure 2.2: Overall pipeline of the proposed framework for 3D multi-person pose estimation from a single RGB image. The proposed framework can recover the absolute camera-centered coordinates of multiple persons' keypoints.

localized keypoints to a certain human. Kocabas *et al.* [46] proposed a pose residual network for assigning detected keypoints to each person.

3D single-person pose estimation. Current 3D single-person pose estimation methods can be categorized into single- and two-stage approaches. The single-stage approach directly localizes the 3D body keypoints from the input image. The two-stage methods utilize the high accuracy of 2D human pose estimation. They initially localize body keypoints in a 2D space and lift them to a 3D space.

[28–30, 47, 48] are based on the single-stage approach. Li *et al.* [47] proposed a multi-task framework that jointly trains both the pose regression and body part detectors. Tekin *et al.* [48] modeled high-dimensional joint dependencies by adopting an auto-encoder structure. Pavlakos *et al.* [29] extended the U-net shaped network to estimate a 3D heatmap for each joint. They used a coarse-to-fine approach to boost performance. Sun *et al.* [30] introduced compositional loss to consider the joint connection structure. Sun *et al.* [28] used soft-argmax operation to obtain the 3D coordinates of body joints in a differentiable manner.

[31–33,49–52] are based on the two-stage approach. Park *et al.* [49] estimated the initial 2D pose and utilized it to regress the 3D pose. Martinez *et al.* [31] proposed a simple network that directly regresses the 3D coordinates of body joints from 2D

coordinates. Zhou *et al.* [32] proposed a geometric loss to facilitate weakly supervised learning of the depth regression module with images in the wild. Yang *et al.* [33] utilized adversarial loss to handle the 3D human pose estimation in the wild.

3D multi-person pose estimation. Few studies have been conducted on 3D multi-person pose estimation from a single RGB image. Rogez *et al.* [13] proposed a top-down approach called LCR-Net, which consists of localization, classification, and regression parts. The localization part detects a human from an input image, and the classification part classifies the detected human into several anchor-poses. The anchor-pose is defined as a pair of 2D and root-relative 3D pose. It is generated by clustering poses in the training set. Then, the regression part refines the anchor-poses. Mehta *et al.* [3] proposed a bottom-up approach system. They introduced an occlusion-robust pose-map formulation that supports pose inference for more than one person through PAFs [44].

3D human root localization in 3D multi-person pose estimation. Rogez *et al.* [13] estimated both the 2D pose in the image coordinate space and the 3D pose in the camera-centered coordinate space simultaneously. They obtained the 3D location of the human root by minimizing the distance between the estimated 2D pose and projected 3D pose, similar to what Mehta *et al.* [14] did. However, this strategy cannot be generalized to other 3D human pose estimation methods because it requires both the 2D and 3D estimations. For example, many works [28, 29, 32, 33] estimate the 2D image coordinates and root-relative depth values of keypoints. As their methods do not output root-relative camera-centered coordinates of keypoints, such a distance minimization strategy cannot be used. Moreover, contextual information cannot be exploited because the image feature is not considered. For example, it cannot distinguish between a child close to the camera and an adult far from the

camera because their scales in the 2D image are similar.

2.3 Overview of the proposed model

The goal of our system is to recover the absolute camera-centered coordinates of multiple persons' keypoints $\{\mathbf{P}_j^{abs}\}_{j=1}^{J}$, where J denotes the number of joints. To address this problem, we construct our system based on the top-down approach that consists of DetectNet, RootNet, and PoseNet. The DetectNet detects a human bounding box of each person in the input image. The RootNet takes the cropped human image from the DetectNet and localizes the root of the human $\mathbf{R} = (x_R, y_R, Z_R)$, in which x_R and y_R are pixel coordinates, and Z_R is an absolute depth value. The same cropped human image is fed to the PoseNet, which estimates the root-relative 3D pose $\mathbf{P}_j^{rel} = (x_j, y_j, Z_j^{rel})$, in which x_j and y_j are pixel coordinates in the cropped image space and Z_j^{rel} is root-relative depth value. We convert Z_j^{rel} into Z_j^{abs} by adding Z_R and transform x_j and y_j to the original input image space. Then, the final absolute 3D pose $\{\mathbf{P}_j^{abs}\}_{j=1}^{J}$ is obtained by simple back-projection.

2.4 DetectNet

We use Mask R-CNN [36] as the framework of DetectNet. Mask R-CNN [36] consists of three parts. The first one, backbone, extracts useful local and global features from the input image by using a deep residual network (ResNet) [53] and feature pyramid network [54]. Based on the extracted features, the second part, the region proposal network, proposes human bounding box candidates. The RoIAlign layer extracts the features of each proposal and passes them to the third part, which is the classification head network. The head network determines whether the given proposal is human

or not and estimates the bounding box refinement offsets. It achieves state-of-the-art performance on publicly available object detection datasets [1]. Due to its high performance and publicly available code [55, 56], we use Mask R-CNN [36] as a DetectNet in our pipeline.

2.5 PoseNet

2.5.1 Model design

The PoseNet estimates the root-relative 3D pose $\mathbf{P}_j^{rel} = (x_j, y_j, Z_j^{rel})$ from a cropped human image. Many works have been presented for this topic [14, 28–33]. Among them, we use the model of Sun *et al.* [28], which is the current state-of-the-art method. This model consists of two parts. The first part is the backbone, which extracts a useful global feature from the cropped human image using ResNet [53]. Second, the pose estimation part takes a feature map from the backbone part and upsamples it using three consecutive deconvolutional layers with batch normalization layers [57] and ReLU activation function. A 1-by-1 convolution is applied to the upsampled feature map to produce the 3D heatmaps for each joint. The soft-argmax operation is used to extract the 2D image coordinates (x_j, y_j), and the root-relative depth values Z_j^{rel}.

2.5.2 Loss function

We train the PoseNet by minimizing the $L1$ distance between the estimated and groundtruth coordinates. The loss function L_{pose} is defined as follows:

$$L_{pose} = \frac{1}{J} \sum_{j=1}^{J} \|\mathbf{P}_j^{rel} - \mathbf{P}_j^{rel*}\|_1, \tag{2.1}$$

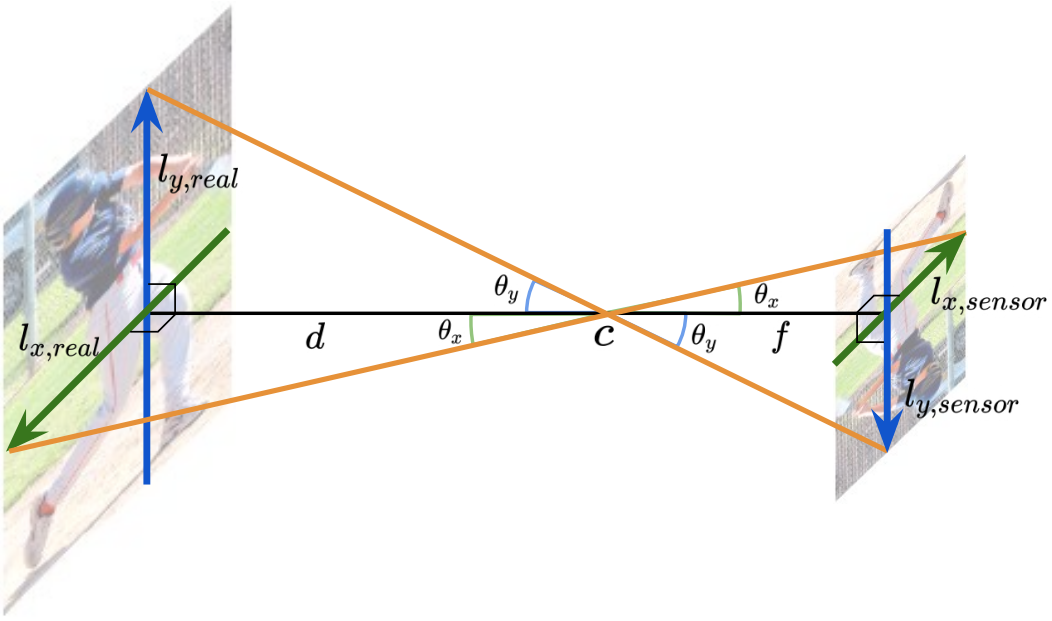


Figure 2.3: Visualization of a pinhole camera model. The green and blue arrows represent the human root joint centered x and y-axes, respectively. The yellow lines show rays, and c is the hole. d, f, and l_{sensor} are distance between camera and the human root joint (mm), focal length (mm), and the length of human on the image sensor (mm), respectively.

where $*$ indicates groundtruth.

2.6 RootNet

2.6.1 Model design

The RootNet estimates the camera-centered coordinates of the human root $\mathbf{R} = (x_R, y_R, Z_R)$ from a cropped human image. To obtain them, RootNet separately estimates the 2D image coordinates (x_R, y_R) and the depth value (*i.e.*, the distance from the camera Z_R) of the human root. The estimated 2D image coordinates are back-projected to the camera-centered coordinate space using the estimated depth value, which becomes the final output.

Considering that an image provides sufficient information on where the human

root is located in the image space, the 2D estimation part can learn to localize it easily. By contrast, estimating the depth only from a cropped human image is difficult because the input does not provide information on the relative position of the camera and human. To resolve this issue, we introduce a new distance measure, k, which is defined as follows:

$$k = \sqrt{\alpha_x \alpha_y \frac{A_{real}}{A_{img}}}, \tag{2.2}$$

where α_x, α_y, A_{real}, and A_{img} are focal lengths divided by the per-pixel distance factors (pixel) of x- and y-axes, the area of the human in real space (mm^2), and image space (pixel^2), respectively. k approximates the absolute depth from the camera to the object using the ratio of the actual area and the imaged area of it, given camera parameters. Eq 2.2 can be easily derived by considering a pinhole camera projection model, as shown in Figure 2.3. The distance d (mm) between the camera and object can be calculated as follows:

$$d = \alpha_x \frac{l_{x,real}}{l_{x,img}} = \alpha_y \frac{l_{y,real}}{l_{y,img}}, \tag{2.3}$$

where $l_{x,real}$, $l_{x,img}$, $l_{y,real}$, $l_{y,img}$ are the lengths of an object in real space (mm) and in image space (pixel), on the x and y-axes, respectively.

By multiplying the two representations of d in Eq 2.3 and taking the square root of it, we can have the 2D extended version of depth measure k in Eq 2.2. Assuming that A_{real} is constant and using α_x and α_y from datasets, the distance between the camera and an object can be measured from the area of the bounding box. As we only consider humans, we assume that A_{real} is $2000mm \times 2000mm$. The area of

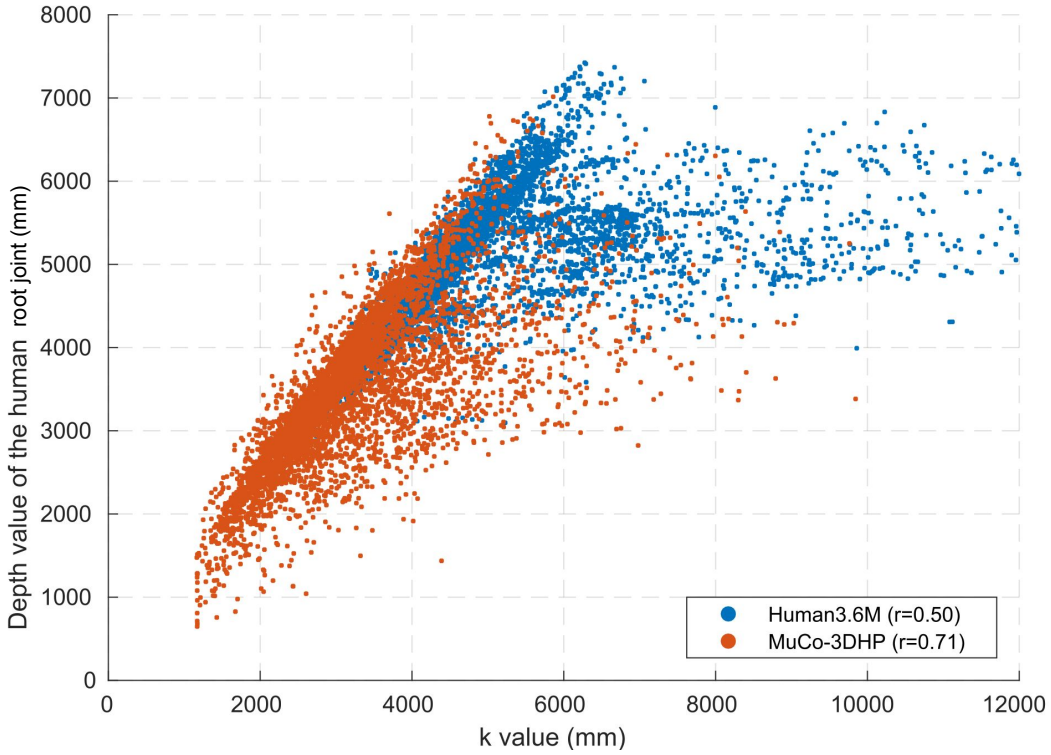


Figure 2.4: Correlation between k and real depth value of the human root. Human3.6M [2] and MuCo-3DHP [3] datasets were used. r represents Pearson correlation coefficient.

the human bounding box is used as A_{img} after extending it to fixed aspect ratio (*i.e.*, height:width = 1:1). Figure 2.4 shows that such an approximation provides a meaningful correlation between k and the real depth values of the human root in 3D human pose estimation datasets [2, 3].

Although k can represent how far the human is from the camera, it can be wrong in several cases because it assumes that A_{img} is an area of A_{real} (*i.e.*, $2000mm \times 2000mm$) in the image space when the distance between the human and the camera is k. However, as A_{img} is obtained by extending the 2D bounding box, it can have a different value according to its appearance, although the distance to the camera is the same. For example, as shown in Figure 2.5(a), two humans have different

Figure 2.5: Examples where k fails to represent the distance between a human and the camera because of incorrect A_{img}.

A_{img} although they are at the same distance to the camera. On the other hand, in some cases, A_{img} can be the same, even with different distances from the camera. For example, in Figure 2.5(b), a child and an adult have similar A_{img}; however, the child is closer to the camera than the adult.

To handle this issue, we design the RootNet to utilize the image feature to correct A_{img}, eventually k. The image feature can give a clue to the RootNet about how much the A_{img} has to be changed. For example, in Figure 2.5(a), the left image can tell the RootNet to increase the area because the human is in a crouching posture. Also, in Figure 2.5(b), the right image can tell the RootNet to increase the area because the input image contains a child. Specifically, the RootNet outputs the correction factor γ from the image feature. The estimated γ is multiplied by the given A_{img}, which becomes A_{img}^{γ}. From A_{img}^{γ}, k is calculated and it becomes the final depth value.

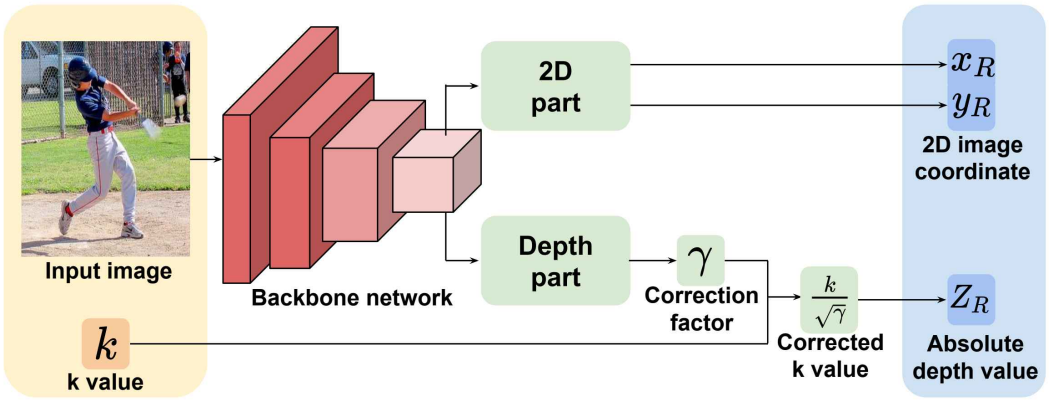


Figure 2.6: Network architecture of the RootNet. The RootNet estimates the 3D human root coordinate.

2.6.2 Camera normalization

Our RootNet outputs correction factor γ only from an input image. Therefore, data from any camera intrinsic parameters (*i.e.*, α_x and α_y) can be used during training and testing. We call this property *camera normalization*, which makes our RootNet very flexible. For example, in the training stage, data from different α_x and α_y can be used together. Also, in the testing stage, RootNet can be used when α_x and α_y are not available, likely for in-the-wild images. In this case, α_x and α_y can be set to any values α'_x and α'_y, respectively. Then, estimated Z_R represents distance between an object and camera whose α_x and α_y are α'_x and α'_y, respectively.

2.6.3 Network architecture

The network architecture of the RootNet, which comprises three components, is visualized in Figure 2.6. First, a backbone network extracts the useful global feature of the input human image using ResNet [53]. Second, the 2D image coordinate estimation part takes a feature map from the backbone part and upsamples it using three consecutive deconvolutional layers with batch normalization layers [57] and

ReLU activation function. Then, a 1-by-1 convolution is applied to produce a 2D heatmap of the root. Soft-argmax [28] extracts 2D image coordinates x_R, y_R from the 2D heatmap. The third component is the depth estimation part. It also takes a feature map from the backbone part and applies global average pooling. Then, the pooled feature map goes through a 1-by-1 convolution, which outputs a single scalar value γ. The final absolute depth value Z_R is obtained by multiplying k with $\frac{1}{\sqrt{\gamma}}$. In practice, we implemented the RootNet to output $\gamma' = \frac{1}{\sqrt{\gamma}}$ directly and multiply it with the k to obtain the absolute depth value Z_R (*i.e.*, $Z_R = \gamma' k$).

2.6.4 Loss function

We train the RootNet by minimizing the $L1$ distance between the estimated and groundtruth coordinates. The loss function L_{root} is defined as follows:

$$L_{root} = \|\mathbf{R} - \mathbf{R}^*\|_1, \tag{2.4}$$

where $*$ indicates the groundtruth.

2.7 Implementation details

Publicly released Mask R-CNN model [56] pre-trained on the COCO dataset [1] is used for the DetectNet without fine-tuning on the human pose estimation datasets [2, 3]. For the RootNet and PoseNet, PyTorch [58] is used for implementation. Their backbone part is initialized with the publicly released ResNet-50 [53] pre-trained on the ImageNet dataset [59], and the weights of the remaining part are initialized by Gaussian distribution with $\sigma = 0.001$. The weights are updated by the Adam optimizer [60] with a mini-batch size of 128. The initial learning rate is set to $1 \times$

10^{-3} and reduced by a factor of 10 at the 17th epoch. We use 256×256 as the size of the input image of the RootNet and PoseNet. We perform data augmentation, including rotation (±30°), horizontal flip, color jittering, and synthetic occlusion [61] in training. Horizontal flip augmentation is performed in testing for the PoseNet following Sun *et al.* [28]. We train the RootNet and PoseNet for 20 epochs with four NVIDIA 1080 Ti GPUs, which took two days, respectively.

2.8 Experiment

2.8.1 Dataset and evaluation metric

Human3.6M dataset. Human3.6M dataset [2] is the largest 3D single-person pose benchmark. It consists of 3.6 million video frames. 11 subjects performing 15 activities are captured from 4 camera viewpoints. The groundtruth 3D poses are obtained using a motion capture system. Two evaluation metrics are widely used. The first one is mean per joint position error (MPJPE) [2], which is calculated after aligning the human root of the estimated and groundtruth 3D poses. The second one is MPJPE after further alignment (*i.e.*, Procrustes analysis (PA) [62]). This metric is called PA MPJPE. To evaluate the localization of the absolute 3D human root, we introduce the mean of the Euclidean distance between the estimated coordinates of the root $\mathbf{R}$ and the groundtruth $\mathbf{R}^*$, *i.e.*, the mean of the root position error (MRPE), as a new metric:

$$MRPE = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{R}^{(i)} - \mathbf{R}^{(i)*}||_2, \tag{2.5}$$

where superscript i is the sample index, and N denotes the total number of test samples.

MuCo-3DHP and MuPoTS-3D datasets. These are the 3D multi-person pose estimation datasets proposed by Mehta *et al.* [3]. The training set, MuCo-3DHP, is generated by compositing the existing MPI-INF-3DHP 3D single-person pose estimation dataset [14]. The test set, MuPoTS-3D dataset, was captured from outdoors, and it includes 20 real-world scenes with groundtruth 3D poses for up to three subjects. The groundtruth is obtained with a multi-view marker-less motion capture system. For evaluation, a 3D percentage of correct keypoints (3DPCK$_{rel}$) and area under 3DPCK curve from various thresholds (AUC$_{rel}$) is used after root alignment with groundtruth. It treats a joint's prediction as correct if it lies within 15cm from the groundtruth joint location. We additionally define 3DPCK$_{abs}$, which is the 3DPCK without root alignment to evaluate the absolute camera-centered coordinates. To evaluate the localization of the absolute 3D human root, we use the average precision of 3D human root location (AP_{25}^{root}), which considers a prediction is correct when the Euclidean distance between the estimated and the groundtruth coordinates is smaller than 25cm.

2.8.2 Experimental protocol

Human3.6M dataset. Two experimental protocols are widely used. *Protocol 1* uses six subjects (S1, S5, S6, S7, S8, S9) in training and S11 in testing. PA MPJPE is used as an evaluation metric. *Protocol 2* uses five subjects (S1, S5, S6, S7, S8) in training and two subjects (S9, S11) in testing. MPJPE is used as an evaluation metric. We use every 5th and 64th frames in videos for training and testing, respectively following [28,30]. When training, besides the Human3.6M dataset, we used additional MPII 2D

Settings	MRPE	MPJPE	Time
Joint learning	138.2	116.7	**0.132**
Disjointed learning (Ours)	**120.0**	**57.3**	0.141

Table 2.1: MRPE, MPJPE, and seconds per frame comparison between joint and disjointed learning on Human3.6M dataset.

human pose estimation dataset [63] following [28–30, 32]. Each mini-batch consists of half Human3.6M and half MPII data. For MPII data, the loss value of the z-axis becomes zero for both of the RootNet and PoseNet following Sun *et al.* [28].

MuCo-3DHP and MuPoTS-3D datasets. Following the previous protocol, we composite 400K frames of which half are background augmented. For augmentation, we use images from the COCO dataset [1] except for images with humans. We use an additional COCO 2D human keypoint detection dataset [1] when training our models on the MuCo-3DHP dataset following Mehta *et al.* [3]. Each mini-batch consists of half MuCo-3DHP and half COCO data. For COCO data, the loss value of z-axis becomes zero for both of the RootNet and PoseNet following Sun *et al.* [28].

2.8.3 Ablation study

In this study, we show how each component of our proposed framework affects the 3D multi-person pose estimation accuracy. To evaluate the performance of the DetectNet, we use the average precision of the bounding box (AP^{box}) following metrics of the COCO object detection benchmark [1].

Disjointed pipeline. To demonstrate the effectiveness of the disjointed pipeline (*i.e.*, separated DetectNet, RootNet, and PoseNet), we compare MRPE, MPJPE, and running time of joint and disjointed learning of the RootNet and PoseNet in Table 2.1. The running time includes DetectNet and is measured using a single TitanX Maxwell GPU. For the joint learning, we combine the RootNet and PoseNet

DetectNet	RootNet	AP^{box}	AP^{root}_{25}	AUC_{rel}	3DPCK_{abs}
R-50	k	43.8	5.2	39.2	9.6
R-50	Ours	43.8	28.5	39.8	31.5
X-101-32	Ours	**45.0**	**31.0**	**39.8**	**31.5**
GT	Ours	100.0	31.4	39.8	31.6
GT	GT	100.0	100.0	39.8	80.2

Table 2.2: Overall performance comparison for different DetectNet and RootNet settings on the MuPoTS-3D dataset.

into a single model which shares backbone part (*i.e.*, ResNet [53]). The image feature from the backbone is fed to each branch of RootNet and PoseNet in a parallel way. Compared with the joint learning, our disjointed learning gives lower error under a similar running time. We believe that this is because each task of RootNet and PoseNet is not highly correlated; therefore, jointly training all tasks can make training harder, resulting in lower accuracy.

Effect of the DetectNet. To show how the performance of the human detection affects the accuracy of the final 3D human root localization and 3D multi-person pose estimation, we compare AP^{root}_{25}, AUC_{rel}, and 3DPCK_{abs} using the DetectNet in various backbones (*i.e.*, ResNet-50 [53], ResNeXt-101-32 [64]) and groundtruth box in the second, third, and fourth row of Table 2.2, respectively. The table shows that based on the same RootNet (*i.e.*, Ours), a better human detection model improves both the 3D human root localization and 3D multi-person pose estimation performance. However, the groundtruth box does not improve overall accuracy considerably compared with other DetectNet models. Therefore, we have sufficient reasons to believe that the given boxes cover most of the person instances with such a high detection AP. We can also conclude that the bounding box estimation accuracy does not have a large impact on the 3D multi-person pose estimation accuracy.

Effect of the RootNet. To show how the performance of the 3D human root lo-

calization affects the accuracy of the 3D multi-person pose estimation, we compare AUC_{rel} and 3DPCK_{abs} using various RootNet settings in Table 2.2. The first and second rows show that based on the same DetectNet (*i.e.*, R-50), our RootNet exhibits significantly higher AP_{25}^{root} and 3DPCK_{abs} compared with the setting in which k is directly utilized as a depth value. We use the x and y of the RootNet when the k is used as a depth value. This result demonstrates that the RootNet successfully corrects the k value. The fourth and last rows show that the groundtruth human root provides similar AUC_{rel}, but significantly higher 3DPCK_{abs} compared with our RootNet. This finding shows that better human root localization is required to achieve more accurate absolute 3D multi-person pose estimation results.

Effect of the PoseNet. All settings in Table 2.2 provides similar AUC_{rel}. Especially, the first and last rows of the table show that using groundtruth box and human root does not provide significantly higher AUC_{rel}. As the results in the table are based on the same PoseNet, we can conclude that AUC_{rel}, which is an evaluation of the root-relative 3D human pose estimation, highly depends on the accuracy of the PoseNet.

2.8.4 Comparison with state-of-the-art methods

Human3.6M dataset. We compare 3D human pose estimation error between ours and state-of-the-art methods on the Human3.6M dataset [2] in Tables 2.3 and 2.4. As most of the previous methods use the groundtruth information (*i.e.*, bounding boxes or 3D root locations) in inference time, we report the performance of the PoseNet using the groundtruth 3D root location. Note that our full model does not require any groundtruth information in inference time. The tables show that our method achieves comparable performance despite not using any groundtruth

Methods	Dir.	Dis.	Eat	Gre.	Phon.	Pose	Pur.	Sit	SitD.	Smo.	Phot.	Wait	Walk	WalkD.	WalkP.	Avg
With groundtruth information in inference time																
Yasin [65]	88.4	72.5	108.5	110.2	97.1	81.6	107.2	119.0	170.8	108.2	142.5	86.9	92.1	165.7	102.0	108.3
Chen [50]	71.6	66.6	74.7	79.1	70.1	67.6	89.3	90.7	195.6	83.5	93.3	71.2	55.7	85.9	62.5	82.7
Moreno [66]	67.4	63.8	87.2	73.9	71.5	69.9	65.1	71.7	98.6	81.3	93.3	74.6	76.5	77.7	74.6	76.5
Zhou [67]	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1	53.7	65.5	51.6	50.4	54.8	55.9	55.3
Martinez [31]	39.5	43.2	46.4	47.0	51.0	41.4	40.6	56.5	69.4	49.2	56.0	45.0	38.0	49.5	43.1	47.7
Sun [30]	42.1	44.3	45.0	45.4	51.5	43.2	41.3	59.3	73.3	51.0	53.0	44.0	38.3	48.0	44.8	48.3
Fang [51]	38.2	41.7	43.7	44.9	48.5	40.2	38.2	54.5	64.4	47.2	55.3	44.3	36.7	47.3	41.7	45.7
Sun [28]	36.9	36.2	40.6	40.4	41.9	34.9	35.7	50.1	59.4	40.4	44.9	39.0	30.8	39.8	36.7	40.6
Ours (PoseNet)	**31.0**	**30.6**	**39.9**	**35.5**	**34.8**	**30.2**	**32.1**	**35.0**	**43.8**	**35.7**	**37.6**	**30.1**	**24.6**	**35.7**	**29.3**	**34.0**
Without groundtruth information in inference time																
Rogez [68]*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	42.7
Ours (Full)	**32.5**	**31.5**	**41.5**	**36.7**	**36.3**	**31.9**	**33.2**	**36.5**	**44.4**	**36.7**	**38.7**	**31.2**	**25.6**	**37.1**	**30.5**	**35.2**

Table 2.3: PA MPJPE comparison with state-of-the-art methods on the Human3.6M dataset using Protocol 1. $*$ used extra synthetic data for training.

Methods	Dir.	Dis.	Eat	Gre.	Phon.	Pose	Pur.	Sit	SitD.	Smo.	Phot.	Wait	Walk	WalkD.	WalkP.	Avg
With groundtruth information in inference time																
Chen [50]	89.9	97.6	90.0	107.9	107.3	93.6	136.1	133.1	240.1	106.7	139.2	106.2	87.0	114.1	90.6	114.2
Tome [69]	65.0	73.5	76.8	86.4	86.3	68.9	74.8	110.2	173.9	85.0	110.7	85.8	71.4	86.3	73.1	88.4
Moreno [66]	69.5	80.2	78.2	87.0	100.8	76.0	69.7	104.7	113.9	89.7	102.7	98.5	79.2	82.4	77.2	87.3
Zhou [67]	68.7	74.8	67.8	76.4	76.3	84.0	70.2	88.0	113.8	78.0	98.4	90.1	62.6	75.1	73.6	79.9
Mehta [14]	57.5	68.6	59.6	67.3	78.1	56.9	69.1	98.0	117.5	69.5	82.4	68.0	55.3	76.5	61.4	72.9
Martinez [31]	51.8	56.2	58.1	59.0	69.5	55.2	58.1	74.0	94.6	62.3	78.4	59.1	49.5	65.1	52.4	62.9
Fang [51]	50.1	54.3	57.0	57.1	66.6	53.4	55.7	72.8	88.6	60.3	73.3	57.7	47.5	62.7	50.6	60.4
Sun [30]	52.8	54.8	54.2	54.3	61.8	53.1	53.6	71.7	86.7	61.5	67.2	53.4	47.1	61.6	63.4	59.1
Sun [28]	**47.5**	**47.7**	**49.5**	**50.2**	**51.4**	**43.8**	**46.4**	**58.9**	**65.7**	**49.4**	**55.8**	**47.8**	**38.9**	**49.0**	**43.8**	**49.6**
Ours (PoseNet)	50.5	55.7	50.1	51.7	53.9	46.8	50.0	61.9	68.0	52.5	55.9	49.9	41.8	56.1	46.9	53.3
Without groundtruth information in inference time																
Rogez [13]	76.2	80.2	75.8	83.3	92.2	79.9	71.7	105.9	127.1	88.0	105.7	83.7	64.9	86.6	84.0	87.7
Mehta [3]	58.2	67.3	61.2	65.7	75.8	62.2	64.6	82.0	93.0	68.8	84.5	65.1	57.6	72.0	63.6	69.9
Rogez [68]*	55.9	60.0	64.5	56.3	67.4	71.8	55.1	**55.3**	84.8	90.7	67.9	57.5	47.8	63.3	54.6	63.5
Mehta [27]	**50.2**	61.9	58.3	58.2	68.8	54.1	61.5	76.8	91.7	63.4	74.6	58.5	48.3	65.3	53.2	63.6
Ours (Full)	51.5	**56.8**	**51.2**	**52.2**	**55.2**	**47.7**	**50.9**	63.3	**69.9**	**54.2**	**57.4**	**50.4**	**42.5**	**57.5**	**47.7**	**54.4**

Table 2.4: MPJPE comparison with state-of-the-art methods on the Human3.6M dataset using Protocol 2. $*$ used extra synthetic data for training.

Methods	MRPE	MRPE_x	MRPE_y	MRPE_z
Baseline [13, 14]	267.8	27.5	28.3	261.9
W/o limb joints	226.2	24.5	24.9	220.2
RANSAC	213.1	24.3	24.3	207.1
RootNet (Ours)	**120.0**	**23.3**	**23.0**	**108.1**

Table 2.5: MRPE comparisons between previous distance minimization-based approaches [13, 14] and our RootNet on the Human3.6M dataset. MRPE_x, MRPE_y, and MRPE_z represent the mean of the errors in the x, y, and z axes, respectively.

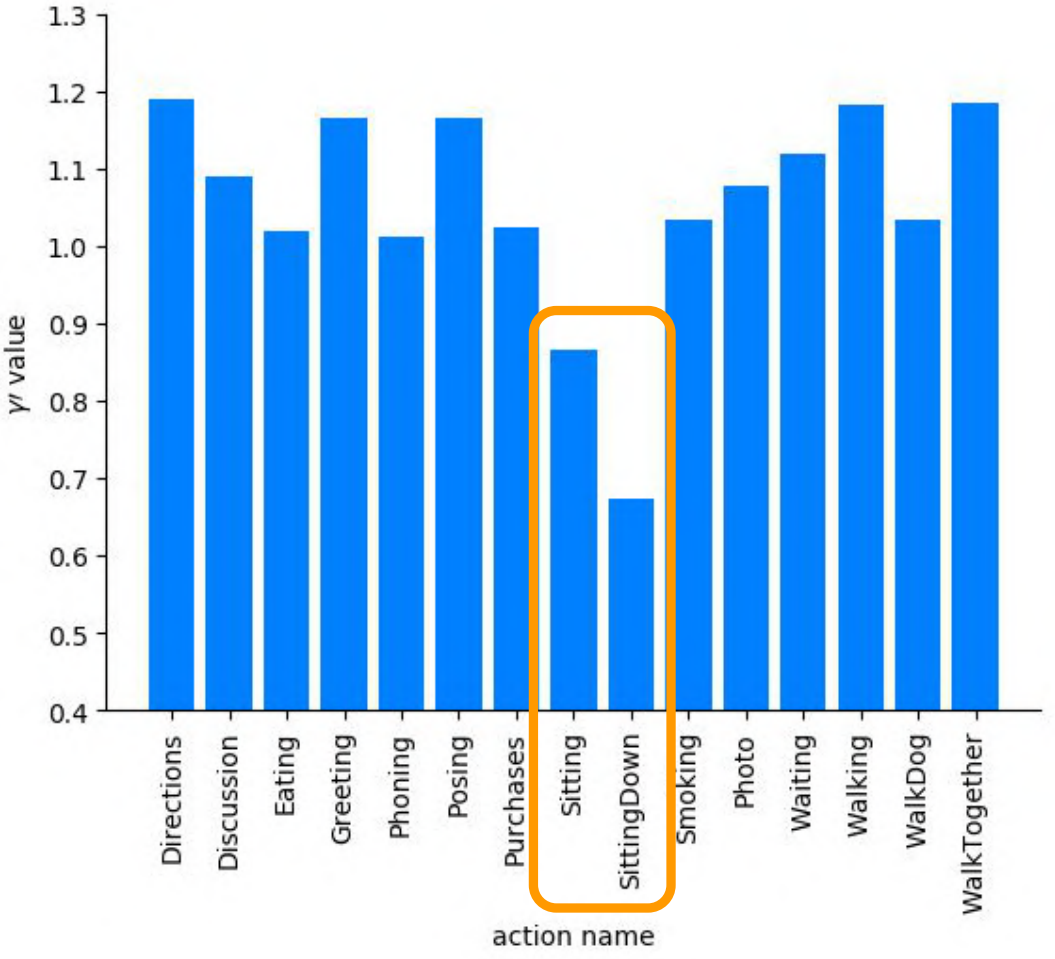


Figure 2.7: Predicted correction factor γ' on Human3.6M dataset. Orange box indicates actions with crouching poses.

information in inference time. Moreover, it significantly outperforms previous 3D multi-person pose estimation methods [1, 3].

In addition, we compare 3D human root joint localization error between ours and previous approaches [13, 14] in Table 2.5. The table shows that our RootNet significantly outperforms previous approaches. Furthermore, the RootNet can be designed independently of the PoseNet, giving design flexibility to both models. In contrast, the previous 3D root localization methods [13, 14] require both 2D and 3D predictions for the root localization, which results in a lack of generalizability. Figure 2.7 shows predicted correction γ' has smaller value when a person in the input image is taking crouching poses (*e.g.*, sitting and sitting down). The figure demonstrates our RootNet adjust the initial depth value k using the image feature.

The previous 3D human root joint localization approaches [13,14] simultaneously estimate 2D image coordinates and 3D camera-centered root-relative coordinates of keypoints. Then, absolute camera-centered coordinates of the human root are obtained by minimizing the distance between 2D predictions and projected 3D predictions. For optimization, the linear least-squares formulation is used. To measure the errors of their method, we implemented and used ResNet-152-based model of Sun *et al.* [28] as a 2D pose estimator and model of Martinez *et al.* [31] as a 3D pose estimator, which are state-of-the-art methods. In addition, to minimize the effect of outliers in 3D-to-2D fitting, we excluded limb joints when fitting. Also, we performed RANSAC with a various number of joints to get optimal joint set for fitting instead of using a heuristically selected joint set.

MuCo-3DHP and MuPoTS-3D datasets. We compare our proposed system with the state-of-the-art 3D multi-person pose estimation methods on the MuPoTS-3D dataset [3] in Tables 2.6 and 2.7. The proposed system significantly outperforms

Methods	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	Avg
Accuracy for all groundtruths																					
Rogez [13]	67.7	49.8	53.4	59.1	67.5	22.8	43.7	49.9	31.1	78.1	50.2	51.0	51.6	49.3	56.2	66.5	65.2	62.9	66.1	59.1	53.8
Mehta [3]	81.0	60.9	64.4	63.0	69.1	30.3	65.0	59.6	64.1	83.9	68.0	68.6	62.3	59.2	70.1	80.0	79.6	67.3	66.6	67.2	66.0
Rogez [68]*	87.3	61.9	67.9	74.6	78.8	48.9	58.3	59.7	78.1	89.5	69.2	73.8	66.2	56.0	74.1	82.1	78.1	72.6	73.1	61.0	70.6
Mehta [27]	88.4	65.1	68.2	72.5	76.2	46.2	65.8	64.1	75.1	82.4	74.1	72.4	64.4	58.8	73.7	80.4	84.3	67.2	74.3	67.8	70.4
Ours	**94.4**	**77.5**	**79.0**	**81.9**	**85.3**	**72.8**	**81.9**	**75.7**	**90.2**	**90.4**	**79.2**	**79.9**	**75.1**	**72.7**	**81.1**	**89.9**	**89.6**	**81.8**	**81.7**	**76.2**	**81.8**
Accuracy only for matched groundtruths																					
Rogez [13]	69.1	67.3	54.6	61.7	74.5	25.2	48.4	63.3	69.0	78.1	53.8	52.2	60.5	60.9	59.1	70.5	76.0	70.0	77.1	81.4	62.4
Mehta [3]	81.0	65.3	64.6	63.9	75.0	30.3	65.1	61.1	64.1	83.9	72.4	69.9	71.0	72.9	71.3	83.6	79.6	73.5	78.9	**90.9**	70.8
Rogez [68]*	88.0	73.3	67.9	74.6	81.8	50.1	60.6	60.8	78.2	89.5	70.8	74.4	72.8	64.5	74.2	84.9	85.2	78.4	75.8	74.4	74.0
Mehta [27]	88.4	70.4	68.3	73.6	82.4	46.4	66.1	83.4	75.1	82.4	76.5	73.0	72.4	73.8	74.0	83.6	84.3	73.9	85.7	90.6	75.8
Ours	**94.4**	**78.6**	**79.0**	**82.1**	**86.6**	**72.8**	**81.9**	**75.8**	**90.2**	**90.4**	**79.4**	**79.9**	**75.3**	**81.0**	**81.0**	**90.7**	**89.6**	**83.1**	**81.7**	77.3	**82.5**

Table 2.6: Sequence-wise 3DPCK$_{rel}$ comparison with state-of-the-art methods on the MuPoTS-3D dataset. $*$ used extra synthetic data for training.

Methods	Hd.	Nck.	Sho.	Elb.	Wri.	Hip	Kn.	Ank.	Avg
Rogez [13]	49.4	67.4	57.1	51.4	41.3	84.6	56.3	36.3	53.8
Mehta [3]	62.1	81.2	77.9	57.7	47.2	**97.3**	66.3	47.6	66.0
Ours	**79.1**	**92.6**	**85.1**	**79.4**	**67.0**	96.6	**85.7**	**73.1**	**81.8**

Table 2.7: Joint-wise 3DPCK$_{rel}$ comparison with state-of-the-art methods on the MuPoTS-3D dataset. All groundtruths are used for evaluation.

DetectNet	RootNet	PoseNet	**Total**
0.120	0.010	0.011	**0.141**

Table 2.8: Seconds per frame for each component of our framework.

them in most of the test sequences and joints.

2.8.5 Running time of the proposed framework

In Table 2.8, we report seconds per frame for each component of our framework. The running time is measured using a single TitanX Maxwell GPU, and the batch size is set to 1. The testing pipeline of the DetectNet is identical to the original Mask R-CNN [36], which resizes the smallest side of the input image to 800 pixels. As the table shows, most of the running time is consumed by DetectNet. It is hard to directly compare running time with previous works [13, 14] because they did not report it. However, we guess that there would be no big difference because models of [13] and [14] are similar with [70] and [44] whose speed is 0.2 and 0.11 seconds per frame, respectively.

2.8.6 Qualitative results

Figures 2.8 and 2.9 show qualitative results of our 3D multi-person pose estimation framework on the MuPoTS-3D [3] and COCO [1] datasets, respectively. Note that COCO dataset consists of *in-the-wild* images which are hardly included in the 3D human pose estimation training sets [2, 3].

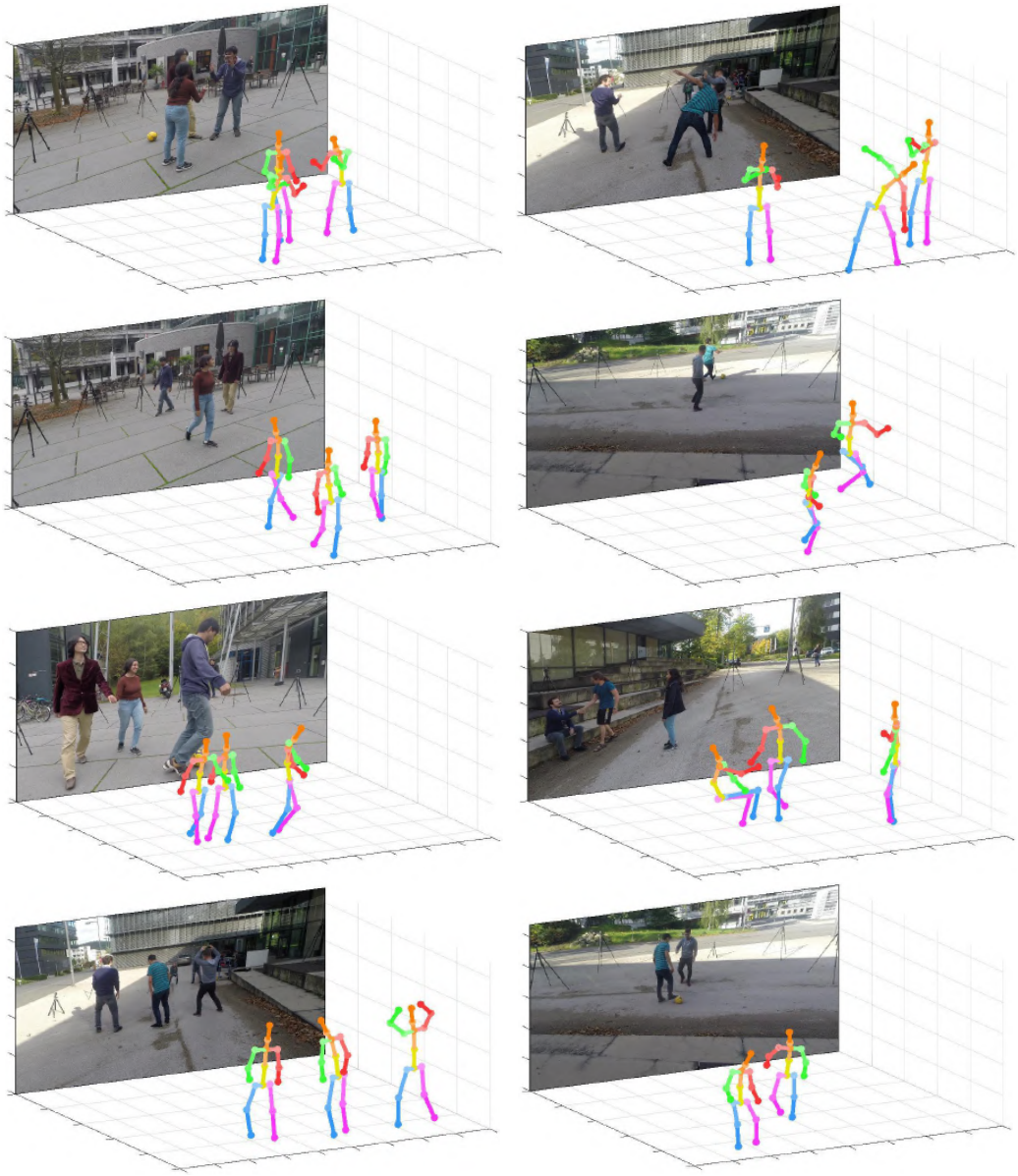


Figure 2.8: Qualitative results of applying our method on the MuPoTS-3D dataset [3].

Figure 2.9: Qualitative results of applying our method on the COCO 2017 [1] validation set.

2.9 Conclusion

We propose a novel and general framework for 3D multi-person pose estimation from a single RGB image. Our framework consists of human detection, 3D human root localization, and root-relative 3D single-person pose estimation models. Since any existing human detection and 3D single-person pose estimation models can be plugged into our framework, it is very flexible and easy to use. The proposed system outperforms previous 3D multi-person pose estimation methods by a large margin and achieves comparable performance with 3D single-person pose estimation methods without any groundtruth information while they use it in inference time. To the best of our knowledge, this work is the first to propose a fully learning-based camera distance-aware top-down approach whose components are compatible with most of the previous human detection and 3D human pose estimation models. We hope that this study provides a new basis for 3D multi-person pose estimation, which has only barely been explored.

Chapter 3

3D Multi-Person Pose and Shape Estimation

3.1 Introduction

3D human pose and mesh estimation aims to simultaneously recover 3D semantic human joint and 3D human mesh vertex locations. Unlike the 3D human pose estimation method of Chapter 2 that only predicts 3D locations of human joints, this chapter aims to predict 3D locations of human mesh vertices additionally. This is a very challenging task because of complicated human articulation and 2D-to-3D ambiguity. It can be used in many applications such as virtual/augmented reality and human action recognition.

SMPL [71] and MANO [72] are the most widely used parametric human body and hand mesh models, respectively, which can represent various human poses and identities. They produce 3D human joint and mesh coordinates from pose and identity parameters. Recent deep convolutional neural network (CNN)-based studies [23–25]

Figure 3.1: Qualitative results of the proposed I2L-MeshNet on MSCOCO [1] and FreiHAND [4] datasets.

for the 3D human pose and mesh estimation are based on the model-based approach, which trains a network to estimate SMPL/MANO parameters from an input image. On the other hand, there have been few methods based on model-free approach [5, 73], which estimates mesh vertex coordinates directly. They obtain the 3D pose by multiplying a joint regression matrix, included in the human mesh model, to the estimated mesh.

Although the recent deep CNN-based methods perform impressive, when estimating the target (*i.e.*, SMPL/MANO parameters or mesh vertex coordinates), all of the previous 3D human pose and mesh estimation works break the spatial relationship among pixels in the input image because of the fully-connected layers at the output stage. In addition, their target representations cannot model the uncertainty of the prediction. The above limitations can make training harder, and as a result, reduce the test accuracy as addressed in [35, 74]. To address the limitations, recent state-of-the-art 3D human pose estimation methods [28, 34, 35], which localize 3D

human joint coordinates without mesh vertex coordinates, utilize the *heatmap* as the target representation of their networks. Each value of one heatmap represents the likelihood of the existence of a human joint at the corresponding pixel positions of the input image and discretized depth value. Therefore, it preserves the spatial relationship between pixels in the input image and models the prediction uncertainty.

Inspired by the recent state-of-the-art heatmap-based 3D human pose estimation methods, we propose I2L-MeshNet, an image-to-lixel prediction network that naturally extends heatmap-based 3D human pose to heatmap-based 3D human pose and mesh. Likewise voxel (volume+pixel) is defined as a quantized cell in three-dimensional space, we define *lixel (line+pixel)* as a quantized cell in one-dimensional space. Our I2L-MeshNet estimates per-lixel likelihood on 1D heatmaps for each mesh vertex coordinates; therefore, it is based on the model-free approach. The previous state-of-the-art heatmap-based 3D human pose estimation methods predict the 3D heatmap of each human joint. Unlike the number of human joints, which is around 20, the number of mesh vertex is much larger (*e.g.*, 6980 for SMPL and 776 for MANO). As a result, predicting 3D heatmaps of all mesh vertices becomes computationally infeasible, which is beyond the limit of modern GPU memory. In contrast, the proposed lixel-based 1D heatmap has an efficient memory complexity, which has a linear relationship with the heatmap resolution. Thus, it allows our system to predict heatmaps with sufficient resolution, which is essential for dense mesh vertex localization.

For more accurate 3D human pose and mesh estimation, we design the I2L-MeshNet as a cascaded network architecture, which consists of PoseNet and MeshNet. The PoseNet predicts the lixel-based 1D heatmaps of each 3D human joint coordinate. Then, the MeshNet utilizes the output of the PoseNet as an additional

input along with the image feature to predict the lixel-based 1D heatmaps of each 3D human mesh vertex coordinate. As the locations of the human joints provide coarse but important information about the human mesh vertex locations, utilizing it for 3D mesh estimation is natural and can increase accuracy substantially.

Our I2L-MeshNet outperforms previous 3D human pose and mesh estimation methods on various 3D human pose and mesh benchmark datasets. Figure 3.1 shows 3D human body and hand mesh estimation results on publicly available datasets. In addition, it can be easily extended to the multi-person 3D human pose and mesh estimation using the framework of Moon *et al.* [26], introduced in Chapter 2. We show the multi-person results in the experimental result section.

Our contributions can be summarized as follows.

- We propose I2L-MeshNet, a novel image-to-lixel prediction network for 3D human pose and mesh estimation from a single RGB image. Our system predicts lixel-based 1D heatmap that preserves the spatial relationship in the input image and models the uncertainty of the prediction.

- Our efficient lixel-based 1D heatmap allows our system to predict heatmaps with sufficient resolution, which is essential for dense mesh vertex localization.

- We show that our I2L-MeshNet outperforms previous state-of-the-art methods on various 3D human pose and mesh datasets.

3.2 Related works

3D human body and hand pose and mesh estimation. Most of the current 3D human pose and mesh estimation methods are based on the model-based approach,

which predicts parameters of pre-defined human body and hand mesh models (*i.e.*, SMPL and MANO, respectively). The model-based methods can be trained only from groundtruth human joint coordinates without mesh vertex coordinates because the model parameters are embedded in low dimensional space. Early model-based methods [75] iteratively fit the SMPL parameters to estimated 2D human joint locations. More recent model-based methods regress the body model parameters from an input image using CNN. Kanazawa *et al.* [23] proposed an end-to-end trainable human mesh recovery (HMR) system that uses the adversarial loss to make their output human shape is anatomically plausible. Pavlakos *et al.* [24] used 2D joint heatmaps and silhouette as cues for predicting accurate SMPL parameters. Omran *et al.* [76] proposed a similar system, which exploits human part segmentation as a cue for regressing SMPL parameters. Xu *et al.* [77] used differentiable rendering to supervise human mesh in the 2D image space. Pavlakos *et al.* [78] proposed a system that uses multi-view color consistency to supervise a network using multi-view geometry. Baek *et al.* [79] trained their network to estimate the MANO parameters using a differentiable renderer. Boukhayma *et al.* [15] trained their network that takes a single RGB image and estimates MANO parameters by minimizing the distance of the estimated hand joint locations and groundtruth. Kolotouros *et al.* [25] introduced a self-improving system consists of SMPL parameter regressor and iterative fitting framework [75].

On the other hand, the model-free approach estimates the mesh vertex coordinates directly instead of regressing the model parameters. Due to the recent advancement of the iterative human body and hand model fitting frameworks [4, 6, 75], pseudo-groundtruth mesh vertex annotation on large-scale datasets [1, 2, 4, 80] became available. Those datasets with mesh vertex annotation motivated several

model-free methods that require mesh supervision. Kolotouros *et al.* [5] designed a graph convolutional human mesh regression system. Their graph convolutional network takes a template human mesh in a rest pose as input and outputs mesh vertex coordinates using image feature from ResNet [53]. Ge *et al.* [73] proposed a graph convolution-based network which directly estimates vertices of hand mesh. Recently, Choi *et al.* [22] proposed a graph convolutional network that recovers 3D human pose and mesh from a 2D human pose.

Unlike all the above model-based and model-free 3D human pose and mesh estimation methods, the proposed I2L-MeshNet outputs 3D human pose and mesh by preserving the spatial relationship between pixels in the input image and modeling uncertainty of the prediction. Those two main advantageous are brought by designing the target of our network to the lixel-based 1D heatmap. This can make training much stable, and the system achieves much lower test error.

Heatmap-based 3D human pose estimation. Most of the recent state-of-the-art 2D and 3D human pose estimation methods use heatmap as a prediction target, which preserves the spatial relationship in the input image and models the uncertainty of the prediction. Tompson *et al.* [74] proposed to estimate the Gaussian heatmap instead of directly regressing coordinates of human body joints. Their heatmap representation helps their model to perform 2D human pose estimation more accurate and motivated many heatmap-based 2D human pose methods [18, 38, 81]. Pavlakos *et al.* [29] and Moon *et al.* [35] firstly proposed to use 3D heatmaps as a prediction target for 3D human body pose and 3D hand pose estimation, respectively. Especially, Moon *et al.* [35] demonstrated that under the same setting, changing prediction target from coordinates to heatmap significantly improves the 3D hand pose accuracy while requires much less amount of the learnable parameters.

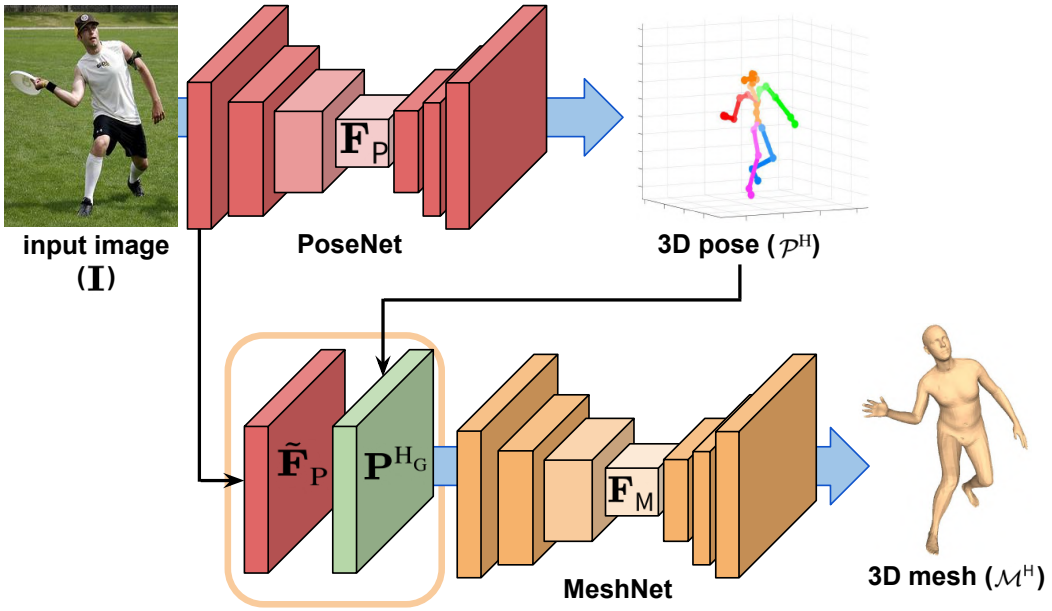


Figure 3.2: Overall pipeline of the proposed I2L-MeshNet.

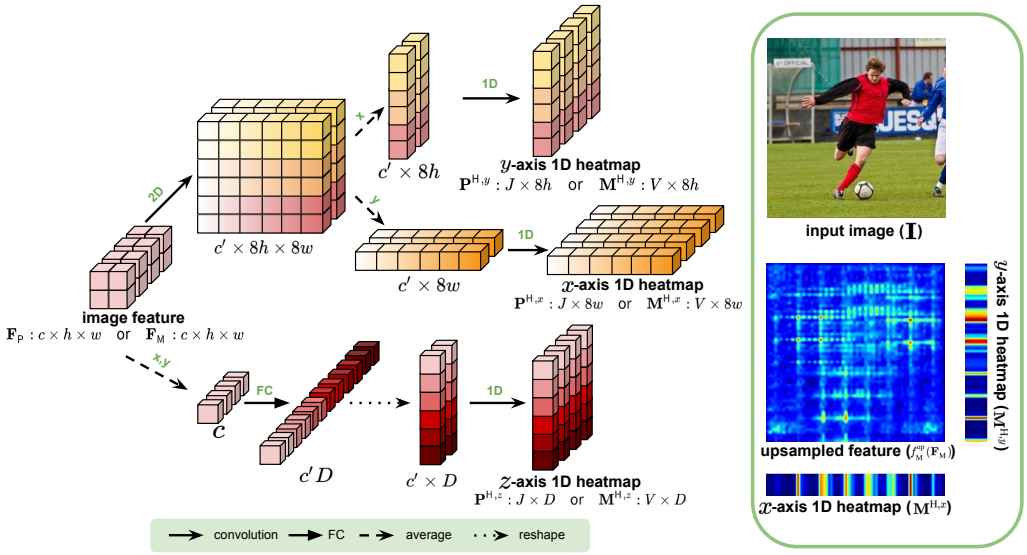
Recently, Moon *et al.* [34] achieved significantly better 3D multi-person pose estimation accuracy using 3D heatmap compared with previous coordinate regression-based methods [13].

3.3 I2L-MeshNet

Figure 3.2 shows the overall pipeline of the proposed I2L-MeshNet. I2L-MeshNet consists of PoseNet and MeshNet, which will be described in the following subsections.

3.3.1 PoseNet

The PoseNet estimates three lixel-based 1D heatmaps of all human joints $\mathcal{P}^{\mathrm{H}} = \{\mathbf{P}^{\mathrm{H},x}, \mathbf{P}^{\mathrm{H},y}, \mathbf{P}^{\mathrm{H},z}\}$ from the input image $\mathbf{I}$. $\mathbf{P}^{\mathrm{H},x}$ and $\mathbf{P}^{\mathrm{H},y}$ are defined in x- and



(a) network architecture to estimate lixel-based 1D heatmaps **(b) visualized feature map and 1D heatmaps**

Figure 3.3: Network architecture to predict lixel-based 1D heatmaps and visualized examples of feature maps and the 1D heatmaps.

y-axis of the image space, while $\mathbf{P}^{\mathrm{H},z}$ is defined in root joint (*i.e.*, pelvis or wrist)-relative depth space. For this, PoseNet extracts image feature $\mathbf{F}_\mathrm{P} \in \mathbb{R}^{c \times h \times w}$ from the input image by ResNet [53]. Then, three upsampling modules increases the spatial size of $\mathbf{F}_\mathrm{P}$ by 8 times, while changing channel dimension from $c = 2048$ to $c' = 256$. Each upsampling module consists of deconvolutional layer, 2D batch normalization layer [57], and ReLU function. The upsampled features are used to compute lixel-based 1D human pose heatmaps, as illustrated in Figure 3.3 (a). We obtain x- and y-axis 1D human pose heatmaps as follows:

$$\mathbf{P}^{\mathrm{H},x} = f_\mathrm{P}^{\mathrm{1D},x}(\mathrm{avg}^{y}(f_\mathrm{P}^{\mathrm{up}}(\mathbf{F}_\mathrm{P}))) \quad \text{and} \quad \mathbf{P}^{\mathrm{H},y} = f_\mathrm{P}^{\mathrm{1D},y}(\mathrm{avg}^{x}(f_\mathrm{P}^{\mathrm{up}}(\mathbf{F}_\mathrm{P}))), \tag{3.1}$$

where $f_{\mathrm{P}}^{\mathrm{up}}$ denotes the three upsampling modules of the PoseNet. avg^{i} and $f_{\mathrm{P}}^{\mathrm{1D},i}$ denote i-axis marginalization by averaging and a 1-by-1 1D convolution that changes channel dimension from c' to J for i-axis 1D human pose heatmap estimation, respectively.

We obtain z-axis 1D human pose heatmaps as follows:

$$\mathbf{P}^{\mathrm{H},z} = f_{\mathrm{P}}^{\mathrm{1D},z}(\psi(f_{\mathrm{P}}(\mathrm{avg}^{x,y}(\mathbf{F}_{\mathrm{P}})))), \tag{3.2}$$

where f_{P} and $\psi\colon \mathbb{R}^{c'D} \rightarrow \mathbb{R}^{c'\times D}$ denote a building block and reshape function, respectively. The building block consists of a fully-connected layer, 1D batch normalization layer, and ReLU function, and it changes the activation size from c to $c'D$. D denotes depth discretization size and is equal to $8h = 8w$. We convert the discretized heatmaps of $\mathcal{P}^{\mathrm{H}}$ to continuous coordinates $\mathbf{P}^{\mathrm{C}} = [\mathbf{p}^{\mathrm{C},x}, \mathbf{p}^{\mathrm{C},y}, \mathbf{p}^{\mathrm{C},z}] \in \mathbb{R}^{J\times 3}$ by soft-argmax [28].

3.3.2 MeshNet

The MeshNet has a similar network architecture with that of the PoseNet. Instead of taking the input image $\mathbf{I}$, MeshNet takes a pre-computed image feature from the PoseNet $\tilde{\mathbf{F}}_{\mathrm{P}}$ and 3D Gaussian heatmap $\mathbf{P}^{\mathrm{H_G}} \in \mathbb{R}^{J\times D\times 8h\times 8w}$. $\tilde{\mathbf{F}}_{\mathrm{P}}$ is the input of the first residual block of the PoseNet whose spatial dimension is $8h \times 8w$. $\mathbf{P}^{\mathrm{H_G}}$ is obtained from $\mathbf{P}^{\mathrm{C}}$ as follows:

$$\mathbf{P}^{\mathrm{H_G}}(j,z,y,x) = \exp\left(-\frac{(x-\mathbf{p}_j^{\mathrm{C},x})^2 + (y-\mathbf{p}_j^{\mathrm{C},y})^2 + (z-\mathbf{p}_j^{\mathrm{C},z})^2}{2\sigma^2}\right), \tag{3.3}$$

where $\mathbf{p}_j^{\mathrm{C},x}$, $\mathbf{p}_j^{\mathrm{C},y}$ and $\mathbf{p}_j^{\mathrm{C},z}$ are jth joint x-, y-, and z-axis coordinates from $\mathbf{P}^{\mathrm{C}}$, respectively. σ is set to 2.5.

From $\mathbf{P}^{\mathrm{H_G}}$ and $\tilde{\mathbf{F}}_{\mathrm{P}}$, we obtain image feature $\mathbf{F}_{\mathrm{M}}$ as follows:

$$\mathbf{F}_{\mathrm{M}} = \mathrm{ResNet}_{\mathrm{M}}(f_{\mathrm{M}}(\psi(\mathbf{P}^{\mathrm{H_G}}) \oplus \tilde{\mathbf{F}}_{\mathrm{P}})), \tag{3.4}$$

where $\psi\colon \mathbb{R}^{J\times D\times 8h\times 8w} \rightarrow \mathbb{R}^{JD\times 8h\times 8w}$ and $\oplus$ denote reshape function and concatenation along the channel dimension, respectively. f_{M} is a convolutional block that consists of a 3-by-3 convolutional layer, 2D batch normalization layer, and ReLU function. It changes the channel dimension of the input to the input channel dimension of the first residual block of the ResNet. $\mathrm{ResNet}_{\mathrm{M}}$ is the ResNet starting from the first residual block.

From the $\mathbf{F}_{\mathrm{M}}$, MeshNet outputs three lixel-based 1D heatmaps of all mesh vertices $\mathcal{M}^{\mathrm{H}} = \{\mathbf{M}^{\mathrm{H},x}, \mathbf{M}^{\mathrm{H},y}, \mathbf{M}^{\mathrm{H},z}\}$ in an exactly the same manner with that of PoseNet, as illustrated in Figure 3.3 (a). Likewise heatmaps of PoseNet, $\mathbf{M}^{\mathrm{H},x}$ and $\mathbf{M}^{\mathrm{H},y}$ are defined in x- and y-axis of the image space, while $\mathbf{M}^{\mathrm{H},z}$ is defined in root joint-relative depth space. We obtain x- and y-axis 1D human mesh heatmaps as follows:

$$\mathbf{M}^{\mathrm{H},x} = f_{\mathrm{M}}^{\mathrm{1D},x}(\mathrm{avg}^{y}(f_{\mathrm{M}}^{\mathrm{up}}(\mathbf{F}_{\mathrm{M}}))) \quad \text{and} \quad \mathbf{M}^{\mathrm{H},y} = f_{\mathrm{M}}^{\mathrm{1D},y}(\mathrm{avg}^{x}(f_{\mathrm{M}}^{\mathrm{up}}(\mathbf{F}_{\mathrm{M}}))), \tag{3.5}$$

where $f_{\mathrm{M}}^{\mathrm{up}}$ denotes the three upsampling modules of the MeshNet. $f_{\mathrm{M}}^{\mathrm{1D},i}$ denote a 1-by-1 1D convolution that changes channel dimension from c' to V for i-axis 1D human mesh heatmap estimation, respectively. Figure 3.3 (b) shows visualized $f_{\mathrm{M}}^{\mathrm{up}}(\mathbf{F}_{\mathrm{M}})$, $\mathbf{M}^{\mathrm{H,x}}$, and $\mathbf{M}^{\mathrm{H,y}}$.

We obtain z-axis 1D human mesh heatmaps as follows:

$$\mathbf{M}^{\mathrm{H},z} = f_{\mathrm{M}}^{\mathrm{1D},z}(\psi(f_{\mathrm{M}}(\mathrm{avg}^{x,y}(\mathbf{F}_{\mathrm{M}})))), \tag{3.6}$$

where f_{M} and $\psi\colon \mathbb{R}^{c'D} \rightarrow \mathbb{R}^{c'\times D}$ denote a building block and reshape function, respectively. The building block consists of a fully-connected layer, 1D batch normalization layer, and ReLU function, and it changes the activation size from c to $c'D$. Likewise we did in the PoseNet, we convert the discretized heatmaps of $\mathcal{M}^{\mathrm{H}}$ to continuous coordinates $\mathbf{M}^{\mathrm{C}} = [\mathbf{m}^{\mathrm{C},x}, \mathbf{m}^{\mathrm{C},y}, \mathbf{m}^{\mathrm{C},z}] \in \mathbb{R}^{V\times 3}$ by soft-argmax [28].

3.3.3 Final 3D human pose and mesh

The final 3D human mesh $\mathbf{M}$ and pose $\mathbf{P}$ are obtained as follows:

$$\mathbf{M} = \Pi(\mathbf{T}^{-1}\mathbf{M}^{\mathrm{C}} + \mathbf{R}) \quad \text{and} \quad \mathbf{P} = \mathcal{J}\mathbf{M}, \tag{3.7}$$

where Π, $\mathbf{T}^{-1}$, and $\mathbf{R} \in \mathbb{R}^{1\times 3}$ denote camera back-projection, inverse affine transformation (*i.e.*, 2D crop and resize), and z-axis offset whose element is a depth of the root joint, respectively. $\mathbf{R}$ is obtained from RootNet [34]. We use normalized camera intrinsic parameters if not available following Moon *et al.* [34]. $\mathcal{J} \in \mathbb{R}^{J\times V}$ is a joint regression matrix defined in SMPL or MANO model.

3.3.4 Loss functions

PoseNet pose loss. To train the PoseNet, we use $L1$ loss function defined as

follows:

$$L_{\text{pose}}^{\text{PoseNet}} = \|\mathbf{P}^{\text{C}} - \mathbf{P}^{\text{C}*}\|_1, \tag{3.8}$$

where $*$ indicates groundtruth. z-axis loss becomes zero if z-axis groundtruth is unavailable.

MeshNet pose loss. To train the MeshNet to predict mesh vertex aligned with body joint locations, we use $L1$ loss function defined as follows:

$$L_{\text{pose}}^{\text{MeshNet}} = \|\mathcal{J}\mathbf{M}^{\text{C}} - \mathbf{P}^{\text{C}*}\|_1, \tag{3.9}$$

where $*$ indicates groundtruth. z-axis loss becomes zero if z-axis groundtruth is unavailable.

Mesh vertex loss. To train the MeshNet to output mesh vertex heatmaps, we use $L1$ loss function defined as follows:

$$L_{\text{vertex}} = \|\mathbf{M}^{\text{C}} - \mathbf{M}^{\text{C}*}\|_1, \tag{3.10}$$

where $*$ indicates groundtruth. z-axis loss becomes zero if z-axis groundtruth is unavailable.

Mesh normal vector loss. Following Wang *et al.* [82], we supervise normal vector of predicted mesh to get visually pleasing mesh result. The $L1$ loss function for normal vector supervision is defined as follows:

$$L_{\text{normal}} = \sum_{f} \sum_{\{i,j\} \subset f} \left| \left\langle \frac{\mathbf{m}_i^{\text{C}} - \mathbf{m}_j^{\text{C}}}{\|\mathbf{m}_i^{\text{C}} - \mathbf{m}_j^{\text{C}}\|_2}, n_f^* \right\rangle \right|, \tag{3.11}$$

where f and n_f indicate a mesh face and unit normal vector of face f, respectively.

$\mathbf{m}_i^{\mathrm{C}}$ and $\mathbf{m}_j^{\mathrm{C}}$ denote ith and jth vertex coordinates of $\mathbf{M}^{\mathrm{C}}$, respectively. n_f^* is computed from $\mathbf{M}^{\mathrm{C}*}$, where $*$ denotes groundtruth. The loss becomes zero if groundtruth 3D mesh is unavailable.

Mesh edge length loss. Following Wang *et al.* [82], we supervise edge length of predicted mesh to get visually pleasing mesh result. The $L1$ loss function for edge length supervision is defined as follows:

$$L_{\mathrm{edge}} = \sum_{f} \sum_{\{i,j\} \subset f} = |\|\mathbf{m}_i^{\mathrm{C}} - \mathbf{m}_j^{\mathrm{C}}\|_2 - \|\mathbf{m}_i^{\mathrm{C}*} - \mathbf{m}_j^{\mathrm{C}*}\|_2|, \tag{3.12}$$

where f and $*$ indicate mesh face and groundtruth, respectively. $\mathbf{m}_i^{\mathrm{C}}$ and $\mathbf{m}_j^{\mathrm{C}}$ denote ith and jth vertex coordinates of $\mathbf{M}^{\mathrm{C}}$, respectively. The loss becomes zero if groundtruth 3D mesh is unavailable.

We train our I2L-MeshNet in an end-to-end manner using all the five loss functions as follows:

$$L = L_{\mathrm{pose}}^{\mathrm{PoseNet}} + L_{\mathrm{pose}}^{\mathrm{MeshNet}} + L_{\mathrm{vertex}} + \lambda L_{\mathrm{normal}} + L_{\mathrm{edge}}, \tag{3.13}$$

where $\lambda = 0.1$ is a weight of L_{normal}. For the stable training, we do not back-propagate gradients before $\mathbf{P}^{\mathrm{H_G}}$.

3.4 Implementation details

PyTorch [58] is used for implementation. The backbone part is initialized with the publicly released ResNet-50 [53] pre-trained on the ImageNet dataset [59], and the weights of the remaining part are initialized by Gaussian distribution with $\sigma = 0.001$.

The weights are updated by the Adam optimizer [60] with a mini-batch size of 48. To crop the human region from the input image, we use groundtruth bounding box in both of training and testing stages following previous works [5, 23, 25]. When the bounding box is not available in the testing stage, we trained and tested Mask R-CNN [36] to get the bounding box. The cropped human image is resized to 256×256, thus $D = 64$ and $h = w = 8$. Data augmentations, including scaling (±25%), rotation (±60°), random horizontal flip, and color jittering (±20%), are performed in training. The initial learning rate is set to 10^{-4} and reduced by a factor of 10 at the 10^{th} epoch. We train our model for 12 epochs with three NVIDIA RTX 2080Ti GPUs, which takes 36 hours for training. Our I2L-MeshNet runs at a speed of 25 frames per second (fps).

3.5 Experiment

3.5.1 Datasets and evaluation metrics

Human3.6M. Human3.6M [2] contains 3.6M video frames with 3D joint coordinate annotations. Because of the license problem, previously used groundtruth SMPL parameters of the Human3.6M are inaccessible. Alternatively, we used SMPLify-X [6] to obtain groundtruth SMPL parameters. MPJPE and PA MPJPE are used for the evaluation [34], which is Euclidean distance (mm) between predicted and groundtruth 3D joint coordinates after root joint alignment and further rigid alignment, respectively.

3DPW. 3DPW [80] contains 60 video sequences captured mostly in outdoor conditions. We use this dataset only for evaluation on its defined test set following

targets	spatial	uncertainty	MPJPE	no. param.	GPU mem.
SMPL param.	✗	✗	100.3	91M	4.3 GB
xyz coord.	✗	✗	114.3	117M	5.4 GB
xyz lixel hm. wo. spatial	✗	✓	92.6	82M	4.5 GB
xyz lixel hm. (ours)	✓	✓	**86.2**	**73M**	**4.6 GB**

Table 3.1: The MPJPE, the number of parameters, and the GPU memory usage comparison between various target representations on Human3.6M.

Kolotouros *et al.* [25]. The same evaluation metrics with Human3.6M (*i.e.*, MPJPE and PA MPJPE) are used, following Kolotouros *et al.* [25].

FreiHAND. FreiHAND [4] contains real-captured 130K training images and 4K test images with MANO pose and shape parameters. The evaluation is performed at an online server. Following Zimmermann *et al.* [4], we report PA MPVPE, PA MPJPE, and F-scores.

MSCOCO. MSCOCO [1] contains large-scale in-the-wild images with 2D bounding box and human joint coordinates annotations. We fit SMPL using SMPLify-X [6] on the groundtruth 2D poses and used the fitted meshes as groundtruth 3D meshes. This dataset is used only for the training.

MuCo-3DHP. MuCo-3DHP [3] is generated by compositing the existing MPI-INF-3DHP 3D [14]. 200K frames are composited, and half of them have augmented backgrounds. We used images of the MSCOCO dataset that do not include humans to augment the backgrounds following Moon *et al.* [34]. This dataset is used only for the training.

3.5.2 Ablation study

All models for the ablation study are trained and tested on Human3.6M. As Human3.6M is the most widely used large-scale benchmark, we believe this dataset is suitable for the ablation study.

Benefit of the heatmap-based mesh estimation. To demonstrate the benefit of the heatmap-based mesh estimation, we compare models with various target representations of the human mesh, such as SMPL parameters, vertex coordinates, and heatmap. Table 3.1 shows MPJPE, the number of parameters, and the GPU memory usage comparison between models with different targets. The table shows that our heatmap-based mesh estimation network achieves the lowest errors while using the smallest number of the parameters and consuming small GPU memory.

The superiority of our heatmap-based mesh estimation network is in two folds. First, it can model the uncertainty of the prediction. To validate this, we trained two models that estimate the camera-centered mesh vertex coordinates directly and estimates lixel-based 1D heatmap of the coordinates using two fully-connected layers. Note that the targets of the two models are the same, but their representations are different. As the first network regresses the coordinates directly, it cannot model the uncertainty on the prediction, while the latter one can because of the heatmap target representation. However, both do not preserve the spatial relationship in the input image because of the global average pooling and the fully-connected layers. As the second and third rows of the table show, modeling uncertainty on the prediction significantly decreases the errors while using a smaller number of parameters. In addition, it achieves lower errors than the SMPL parameter regression model, which is the most widely used target representation but cannot model the uncertainty.

targets	mem. complx.	resolution	MPJPE	GPU mem.
xyz voxel hm.	$\mathcal{O}(VD^3)$	8×8×8	102.8	4.3 GB
		16×16×16	-	OOM
xy pixel hm. + z lixel hm.	$\mathcal{O}(VD^2)$	8×8, 8	97.9	3.5 GB
		32×32, 32	89.4	5.7 GB
		64×64, 64	-	OOM
xyz lixel hm. (ours)	$\mathcal{O}(VD)$	8, 8, 8	100.2	3.4 GB
		32, 32 ,32	94.8	4.0 GB
		64, 64, 64	**86.2**	**4.6 GB**

Table 3.2: The MPJPE and the GPU memory usage comparison between various heatmap representations on Human3.6M.

Second, it preserves the spatial relationship between pixels in the input image. The final model estimates the x- and y-axis heatmaps of each mesh vertex in a fully-convolutional way, thus preserves the spatial relationship. It achieves the best performance with the smallest number of parameters while consuming similar GPU memory usage compared with the SMPL parameter regression method that requires the least amount of GPU memory.

In Table 3.1, all models have the same network architecture with our I2L-MeshNet except for the final output prediction part. We removed PoseNet from all models, and the remaining MeshNet directly estimates targets from the input image $\mathbf{I}$. Except for the last row (ours), all settings output targets using two fully-connected layers. We followed the training details of [23,25] for the SMPL parameter estimation.

Lixel-based vs. pixel-based vs. voxel-based heatmap. To demonstrate the effectiveness of the lixel-based 1D heatmap over other heatmap representations, we train three models that predict lixel-based, pixel-based, and voxel-based heatmap, respectively. We used the same network architecture (*i.e.*, MeshNet of the I2L-

MeshNet) for all settings except for the final prediction part. Their networks directly predict the heatmaps from the input image. x-, y-, and z-axis of each heatmap represents the same coordinates. Table 3.2 shows memory complexity, heatmap resolution, MPJPE, and GPU memory usage comparison between models that predict different target representations of human mesh. The table shows that our lixel-based one achieves the lowest error while consuming small GPU memory usage.

Compared with the pixel-based and voxel-based heatmap, our lixel-based one consumes much less amount of GPU memory under the same resolution. The $8 \times 8 \times 8$ voxel-based heatmap requires similar GPU memory usage with that of $64, 64, 64$ lixel-based one, and we found that enlarging the voxel-based heatmap size from it is not allowed in the current GPU memory limit (*i.e.*, 12 GB). The pixel-based heatmap is more efficient than the voxel-based one; however still much inefficient than our lixel-based one, which makes enlarging from $32 \times 32, 32$ impossible. This inefficient memory usage limits the heatmap resolution; however, we found that the heatmap resolution is critical for dense mesh vertex localization. On the other hand, the memory complexity of our lixel-based heatmap is a linear function with respect to D; thus, we can predict a high-resolution heatmap for each mesh vertex. The memory efficiency will be more important when a high-resolution human mesh model is used.

Under the same resolution, the combination of pixel-based heatmap and lixel-based heatmap achieves the best performance. We think that estimating the voxel-based heatmap involves too many parameters at a single output layer, which makes it produce high errors. In addition, lixel-based heatmap inherently involves spatial ambiguity that arises from marginalizing the 2D feature map to 1D, which can be a possible reason for worse performance than the combined one.

settings	MPJPE	PA MPJPE	GPU mem.
avg on $\mathbf{F}_{\mathrm{M}}$	93.5	64.1	**4.4 GB**
avg on $f_{\mathbf{up}}^{\mathbf{M}}(\mathbf{F}_{\mathbf{M}})$ (ours)	**86.2**	**59.8**	4.6 GB

Table 3.3: The MPJPE, PA MPJPE, and GPU memory usage comparison between various marginalization settings on Human3.6M dataset.

Where to marginalize 2D to 1D. We report how the MPJPE, PA MPJPE, and GPU memory usage change when the marginalization takes place on the ResNet output (*i.e.*, $\mathbf{F}_{\mathrm{P}}$ or $\mathbf{F}_{\mathrm{M}}$), which is the input of the first upsampling module, instead of the output of the last upsampling module (*i.e.*, $f_{\mathrm{up}}^{\mathrm{P}}(\mathbf{F}_{\mathrm{P}})$ or $f_{\mathrm{up}}^{\mathrm{M}}(\mathbf{F}_{\mathrm{M}})$) in Table 3.3. For convenience, we removed PoseNet from our I2L-MeshNet and changed MeshNet to take the input image. The table shows that the early marginalization increases errors while requiring less amount of GPU memory. This is because the marginalized two 1D feature maps can be generated from multiple 2D feature map, which results in spatial ambiguity. To reduce the effect of this spatial ambiguity, we designed our I2L-MeshNet to extract a sufficient amount of 2D information and then apply the marginalization at the last part of the network instead of applying it in the early stage.

When the marginalization is applied on the ResNet output $\mathbf{F}_{\mathrm{M}}$, all 2D layers (*i.e.*, deconvolutional layers and batch normalization layers) in the upsampling modules are converted to the 1D layers. All models are trained on Human3.6M dataset. The z-axis heatmap prediction part is not changed.

How to marginalize 2D to 1D. We report how the MPJPE and PA MPJPE change when different marginalization methods are used in Table 3.4. For convenience, we removed PoseNet from our I2L-MeshNet and changed MeshNet to take the input image. The table shows that our average pooling achieves the lowest errors.

settings	MPJPE	PA MPJPE
max pooling	93.5	64.1
weighted sum	89.4	61.4
avg pooling (ours)	**86.2**	**59.8**

Table 3.4: The MPJPE and PA MPJPE comparison between various marginalization settings on Human3.6M dataset.

settings	resolution	uncertainty in z-axis	MPJPE	GPU mem.
2.5D heatmap [83]	8×8, 8×8	✗	107.4	3.6GB
2.5D heatmap [83]	32×32, 32×32	✗	100.4	8.4GB
lixel-based 1D heatmap	8, 8, 8	✓	100.2	3.4GB
lixel-based 1D heatmap	32, 32, 32	✓	94.8	4.0GB
lixel-based 1D heatmap	64, 64, 64	✓	86.2	4.6GB

Table 3.5: The MPJPE and GPU memory usage comparison between various marginalization settings on Human3.6M dataset.

Compared with the max-pooling that provides the gradients to one-pixel position per one x or y position, our average pooling provides the gradients to all pixel positions, which is much richer ones. We implemented the weighted sum by constructing a convolutional layer whose kernel size is $(8h, 1)$ and $(1, 8w)$ for x- and y-axis lixel-based 1D heatmap prediction, respectively, without padding. The weighted sum provides a lower error than that of the max pooling, however still worse than our average pooling. We believe the large size of a kernel of the convolutional layer (*i.e.*, $(8h, 1)$ and $(1, 8w)$) is hard to be optimized, which results in higher error than ours. For all settings, models are trained on Human3.6M dataset, and the z-axis heatmap prediction part is not changed.

Comparison with previous 2.5D heatmap regression We compare the MPJPE and GPU memory usage between a model that predicts our lixel-based 1D heatmap and a model that predicts the 2.5D heatmap [83] in Table 3.5. The 2.5D heatmap [83] consists of xy heatmap and z heatmap, where xy one is the pixel-based 2D heatmap,

settings	3D pose	MPJPE	PA MPJPE
MeshNet	✗	86.2	59.8
PoseNet+MeshNet (ours)	✓	**81.8**	**58.0**
MeshNet	GT	25.5	17.1

Table 3.6: The MPJPE and PA MPJPE comparison between various network cascading strategies on Human3.6M.

and z one has the same spatial size as that of xy heatmap and contains root joint-relative depth on the activated xy position for all mesh vertices. They predict the depth values on z heatmap, not the likelihood, thus cannot model uncertainty of the z-axis prediction. As the table shows, our lixel-based one achieves significantly lower error under the same resolution while requiring a much smaller amount of GPU memory. We think that this is because the 2.5D heatmap of Iqbal *et al.* [83] cannot model uncertainty of the prediction in z-axis, while ours can. For all settings, models are trained on Human3.6M dataset, and we removed PoseNet and changed MeshNet to take an input image and predict the heatmap.

Benefit of the cascaded PoseNet and MeshNet. To demonstrate the benefit of the cascaded PoseNet and MeshNet, we trained and tested three networks using various network cascading strategy. First, we removed PoseNet from the I2L-MeshNet. The remaining MeshNet directly predicts lixel-based 1D heatmap of each mesh vertex from the input image. Second, we trained I2L-MeshNet, which has cascaded PoseNet and MeshNet architecture. Third, to check the upper bound accuracy with respect to the output of the PoseNet, we fed the groundtruth 3D human pose instead of the output of the PoseNet to the MeshNet in both the training and testing stage. Table 3.6 shows utilizing the output of the PoseNet (the second row) achieves better accuracy compared with using only MeshNet (the first row) to estimate the human mesh. Interestingly, passing the groundtruth 3D human pose to the MeshNet

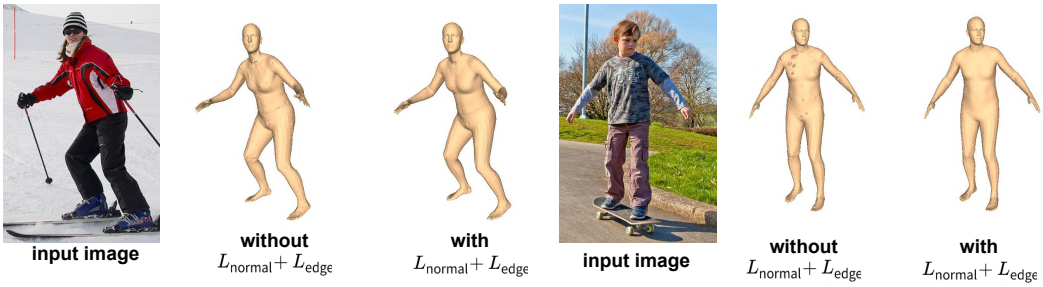


Figure 3.4: Estimated meshes from models trained with different combinations of loss functions.

methods	Human3.6M		3DPW	
	MPJPE	PA MPJPE	MPJPE	PA MPJPE
HMR [23]	153.2	85.5	300.4	137.2
GraphCMR [5]	78.3	59.5	126.5	80.1
SPIN [25]	72.9	51.9	113.1	71.7
I2L-MeshNet (Ours)	**55.7**	**41.7**	**95.4**	**60.8**

Table 3.7: The MPJPE and PA MPJPE comparison on Human3.6M and 3DPW. All methods are trained on Human3.6M and MSCOCO.

(the last row) significantly improves the performance compared with all the other settings. This indicates that improving the 3D human pose estimation network can be one important way to improve 3D human mesh estimation accuracy.

Effect of each loss function For visually pleasant mesh estimation, we use normal vector loss L_{normal} and edge length loss L_{edge}. We show the effectiveness of the two loss functions in Figure 3.4. As the figure shows, the two loss functions improve the visual quality of output meshes. We checked that L_{normal} and L_{edge} marginally affect the MPJPE and PA MPJPE. For all settings, all models are trained on Human3.6M dataset and MSCOCO dataset.

methods	MPJPE	PA MPJPE
SMPLify [75]	-	82.3
Lassner [84]	-	93.9
HMR [23]	88.0	56.8
NBF [76]	-	59.9
Pavlakos [24]	-	75.9
Kanazawa [85]	-	56.9
GraphCMR [5]	-	50.1
Arnab [86]	77.8	54.3
SPIN [25]	-	**41.1**
I2L-MeshNet (Ours)	**55.7**	**41.1**

Table 3.8: The MPJPE and PA MPJPE comparison on Human3.6M. Each method is trained on different datasets.

methods	MPJPE	PA MPJPE
HMR [23]	-	81.3
Kanazawa [85]	-	72.6
GraphCMR [5]	-	70.2
Arnab [86]	-	72.2
SPIN [25]	-	59.2
I2L-MeshNet (Ours)	**93.2**	**57.7**
I2L-MeshNet (Ours) + SMPL regress	100.0	60.0

Table 3.9: The MPJPE and PA MPJPE comparison on 3DPW. Each method is trained on different datasets.

3.5.3 Comparison with state-of-the-art methods

Human3.6M and 3DPW. We compare the MPJPE and PA MPJPE of our I2L-MeshNet with previous state-of-the-art 3D human body pose and mesh estimation methods on Human3.6M and 3DPW test set. As each previous work trained their network on different training sets, we report the 3D errors in two ways.

First, we train all methods on Human3.6M and MSCOCO and report the errors in Table 3.7. The previous state-of-the-art methods [5, 23, 25] are trained from their officially released codes. The table shows that our I2L-MeshNet significantly outperforms previous methods by a large margin on both datasets.

Second, we report the 3D errors of previous methods from their papers and ours in Table 3.8 and Table 3.9. Each network of the previous method is trained on the different combinations of datasets, which include Human3.6M, MSCOCO, MPII [63], LSP [87], LSP-Extended [88], UP [84], and MPI-INF-3DHP [14]. We used MuCo-3DHP for the additional training dataset for the evaluation on the 3DPW dataset. We also report the 3D errors from an additional SMPL parameter regression module following Kolotouros *et al.* [5]. The tables show that the performance gap between ours and the previous state-of-the-art method [25] is significantly reduced.

The reason for the reduced performance gap is that previous model-based state-of-the-art methods [23, 25] can get benefit from many in-the-wild 2D human pose datasets [1, 87, 88] by a 2D pose-based weak supervision. As the human body or hand model assumes a prior distribution between the human model parameters (*i.e.*, 3D joint rotations and identity vector) and 3D joint/mesh coordinates, the 2D pose-based weak supervision can provide gradients in depth axis, calculated from the prior distribution. Although the weak supervision still suffers from the depth ambiguity, utilizing in-the-wild images can be highly beneficial because the images have diverse appearances compared with those of the lab-recorded 3D datasets [2, 3, 14]. On the other hand, model-free approaches, including the proposed I2L-MeshNet, do not assume any prior distribution, therefore hard to get benefit from the weak supervision. Based on the two comparisons, we can draw two important conclusions.

- I2L-MeshNet achieves much higher accuracy than the model-based methods when trained on the same datasets that provide groundtruth 3D human poses and meshes.

- The model-based approaches can achieve comparable or higher accuracy by

methods	PA MPVPE	PA MPJPE	F@5 mm	F@15 mm	GT scale
Hasson *et al.* [16]	13.2	-	0.436	0.908	✓
Boukhayma *et al.* [15]	13.0	-	0.435	0.898	✓
FreiHAND [4]	10.7	-	0.529	0.935	✓
I2L-MeshNet (Ours)	**7.6**	**7.4**	**0.681**	**0.973**	✗

Table 3.10: The PA MPVPE, PA MPJPE, and F-scores comparison between state-of-the-art methods and the proposed I2L-MeshNet on FreiHAND. The checkmark denotes a method use groundtruth information during inference time.

utilizing additional in-the-wild 2D pose data without requiring the 3D supervisions.

We think that a larger number of accurately aligned in-the-wild image-3D mesh data can significantly boost the accuracy of I2L-MeshNet. The iterative fitting [6,75], neural network [89], or their combination [25] can be used to obtain more data. This can be an important future research direction, and we leave this as future work.

FreiHAND. We compare MPVPE and F-scores of our I2L-MeshNet with previous state-of-the-art 3D human hand pose and mesh estimation methods [4, 15, 16]. We trained Mask R-CNN [36] on FreiHAND train images to get the hand bounding box of test images. Table 3.10 shows that the proposed I2L-MeshNet significantly outperforms all previous works without groundtruth scale information during the inference time. We additionally report MPJPE in the table.

MSCOCO. We provide qualitative results comparison between ours and previous state-of-the-art model-free method (*i.e.*, GraphCMR [5]) in Figure 3.5. As the figure shows, our I2L-MeshNet provides much more visually pleasant mesh results than GraphCMR. We think this is because the graph convolutional network (GraphCNN) often tends to smooth the meshes by averaging the vertex feature with that of neighboring vertices.

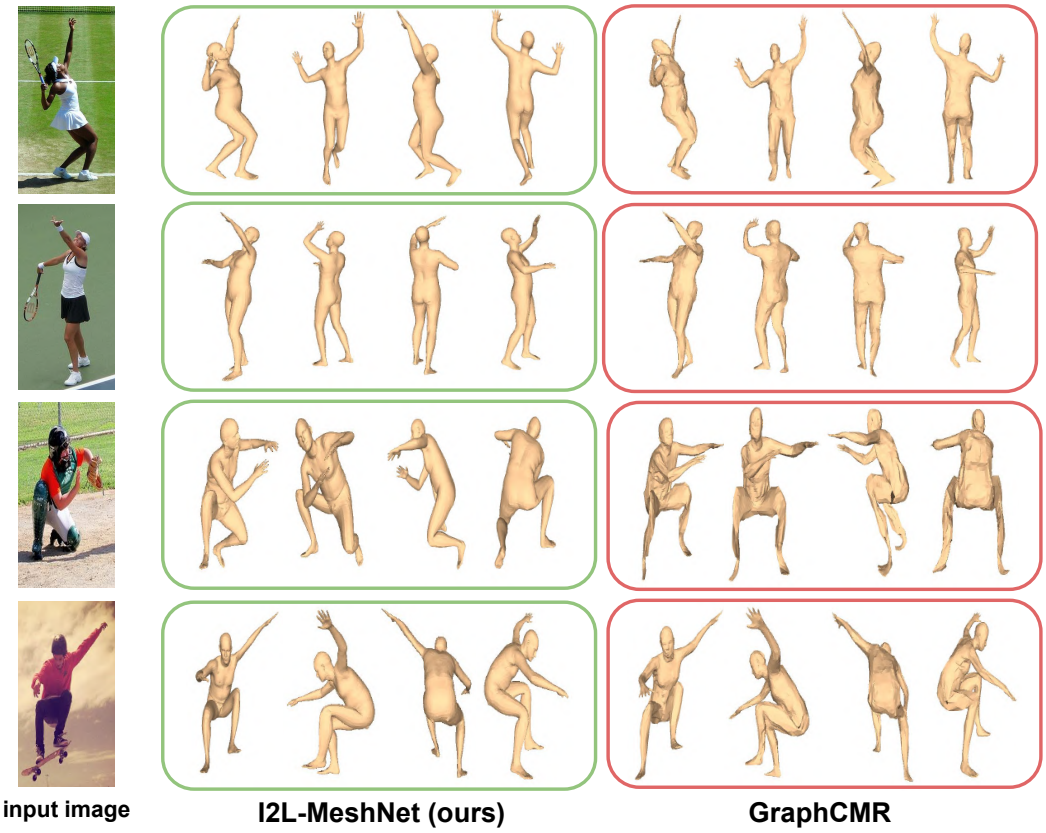


Figure 3.5: Estimated meshes comparisons between our I2L-MeshNet and GraphCMR [5].

Figure 3.6 shows 3D multi-person pose and mesh estimation result of an in-the-wild image of MSCOCO. The framework of Moon *et al.* [26] is used to extend the single person 3D pose and shape of I2L-MeshNet to the multi-person case.

3.6 Conclusion

We propose an I2L-MeshNet, image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. We convert the output of the network to the lixel-based 1D heatmap, which preserves the spatial relation-

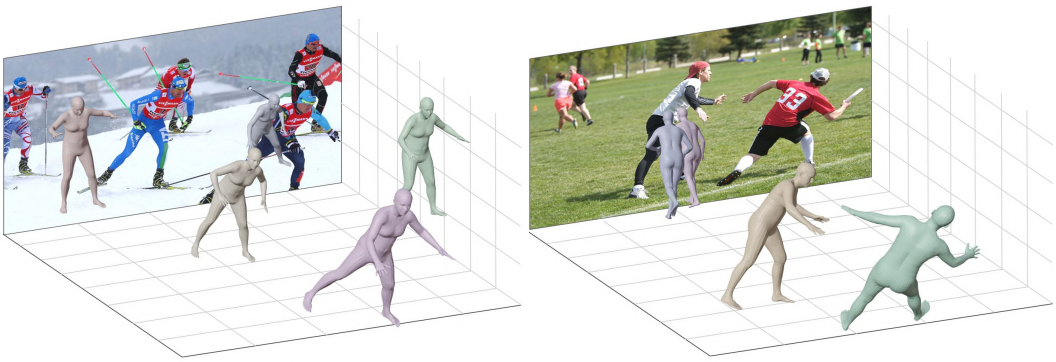


Figure 3.6: 3D multi-person pose and mesh estimation result on an in-the-wild image.

ship in the input image and models uncertainty of the prediction. Our lixel-based 1D heatmap requires much less GPU memory usage under the same heatmap resolution while producing better accuracy compared with a widely used voxel-based 3D heatmap. Our I2L-MeshNet outperforms previous 3D human pose and mesh estimation methods on various 3D human pose and mesh datasets. We hope our method can give useful insight to the following model-free 3D human pose and mesh estimation approaches.

Chapter 4

Expressive 3D Multi-Person Pose and Shape Estimation

4.1 Introduction

Expressive 3D human pose and mesh estimation aims to localize joints and mesh vertices of all human parts, including body, hands, and face, simultaneously in the 3D space. Unlike the 3D human pose and mesh estimation methods of previous chapters are applicable to only one of the body, hands, and face at a time (*i.e.*, part-specific methods), this chapter aims to recover the 3D pose and mesh of the body, hands, and face at the same time (*i.e.*, expressive method). By combining 3D pose and mesh of all human parts, we can understand not only human articulation and shape but also human intention and feeling, which can be useful in motion capture, virtual/augmented reality, and human action recognition. This is a very challenging task and has been addressed by only several recent approaches.

3D rotations of human joints (*i.e.*, 3D rotational pose) represent relative 3D

rotations to a parent joint, defined in a human kinematic chain. Consideration of the 3D rotational pose in addition to the 3D positions of human joints (*i.e.*, 3D positional pose) make human joints 6D object, thus enables skinning functions (*e.g.*, linear blend skinning). As many computer vision and graphics tasks, such as motion capture and animation, are based on skinning functions, they require a 3D rotational pose as an input. Therefore, many previous 3D human pose and mesh estimation methods have been proposed to predict the 3D rotational pose accurately.

Previous 3D human pose and mesh estimation methods [4,5,15,16,23–25] mostly rely on only global image feature to predict 3D rotational pose. They perform global average pooling (GAP) on the extracted image feature from ResNet [53] and pass the pooled feature to several fully connected layers for the 3D rotational pose prediction. The estimated 3D rotations are passed to human model layers (*e.g.*, SMPL [71] for body, MANO [72] for hands, FLAME [90] for face, or SMPL-X [6] for all parts) for the final 3D pose and mesh. Although the global image feature can provide the overall articulation of human, it lacks joint-specific local information, which can be obtained from features on the positional pose. However, GAP in their networks breaks the spatial domain; thus, it limits the chance of utilizing the local features on the positional pose.

To effectively utilize both local and global features, we present *Pose2Pose*, a 3D positional pose-guided 3D rotational pose prediction network. Our Pose2Pose consists of PositionNet and RotationNet. PositionNet predicts the 3D positional pose from an input image in a fully convolutional way. Then, a positional pose-guided pooling extracts joint-specific local and global features on the predicted positional pose of the ResNet output image feature. From the extracted joint-specific features, the RotationNet constructs a human skeleton graph and regresses the 3D rota-

Figure 4.1: Qualitative results of the proposed Pose2Pose on in-the-wild images. Our framework can produce accurate expressive 3D human pose and mesh, which includes body, hands, and face.

tional pose using a joint-specific graph convolution. Unlike the vanilla graph convolution [91] that shares learnable weights for all graph vertices, our joint-specific graph convolution uses separated learnable weights for each joint, which share a similar spirit of Liu *et al.* [92]. This joint-specific graph convolution effectively processes the joint-specific local and global features by learning joint-specific characteristics and different relationships between different joints.

We use our Pose2Pose for expressive 3D human pose and mesh estimation. The proposed Pose2Pose significantly outperforms previous 3D human pose and mesh estimation methods by a large margin. Figure 4.1 shows qualitative results of the proposed Pose2Pose. In addition, it can be easily extended to the multi-person 3D human pose and mesh estimation using the framework of Moon *et al.* [26], introduced in Chapter 2. We show the multi-person results in the experimental result section.

Our contributions can be summarized as follows.

- We present Pose2Pose, a 3D positional pose-guided 3D rotational pose prediction network for expressive 3D human pose and mesh estimation. Our Pose2Pose utilizes joint-specific local and global features, extracted by a positional pose-guided pooling.
- To effectively process joint-specific local and global features, we propose to use a joint-specific graph convolution.
- We show that our Pose2Pose outperforms all previous part-specific and expressive 3D human pose and mesh estimation methods.

4.2 Related works

Expressive 3D human pose and mesh estimation. Due to its difficulty and absence of the unified expressive body model, there have been very few attempts to simultaneously recover the 3D human pose and mesh of all human parts, including body, hands, and face. Most previous attempts are an optimization-based approach, which fits a 3D human model to the 2D/3D evidence. Joo *et al.* [19] fits their human models (*i.e.*, Frank and Adam) to 3D human joints coordinates and point clouds in a multi-view studio environment. Xiang *et al.* [93] extended Joo *et al.* [19] to the single RGB case. Pavlakos *et al.* [6] and Xu *et al.* [94] fits their human model, SMPL-X and GHUM, respectively, to 2D human joint coordinates. As the above optimization-based methods can be slow and prone to noisy evidence, a regression-based approach is presented recently. Choutas *et al.* [7] presented ExPose, which predicts the expressive human pose and mesh using body-driven attention.

Our Pose2Pose is also the regression-based approach; however, it has a clear difference compared with the previous work, ExPose [7]. ExPose consists of body,

hand, and face branches, and each branch relies on only global image features to regress the parameters of the corresponding human model from the input images. On the other hand, our Pose2Pose exploits both local and global features by the positional pose-guided pooling and joint-specific graph convolution. We show that utilizing both local and global features brings significant performance gain.

Local and global features for 3D human pose and mesh estimation. Utilizing both local and global features has been proven to be crucial for accurate 3D human pose and mesh estimation. Detection-based 3D human pose and mesh estimation methods [21, 28, 29, 35] have achieved high 3D positional pose accuracy by utilizing both local and global features. They *detect* the human joints or mesh vertices from an input image by predicting heatmaps, which have activations where human joints or mesh vertices likely exist. As the heatmap is predicted in a fully convolutional way, the detection-based methods do not require GAP; thus, they can utilize both local and global features. However, their methods are hard to be used to predict 3D rotational pose because the input image only contains the position and intensity of each pixel; thus, the 3D rotational pose cannot be predicted in a fully convolutional way.

On the other hand, regression-based methods [5, 23–25] can predict both 3D positional and rotational pose by a direct *regression*. However, previous regression-based methods suffer from low accuracy compared with detection-based methods because the previous ones rely only on global features, obtained by GAP. Several works attempted to utilize both local and global features. Guler and Kokkinos [95] and Zhang *et al.* [96] extract joint-specific local and global features; however, they did not pass the predicted positional pose to the final 3D rotational pose prediction module. Especially, Zhang *et al.* [96] predicts 2D positional pose, which cannot

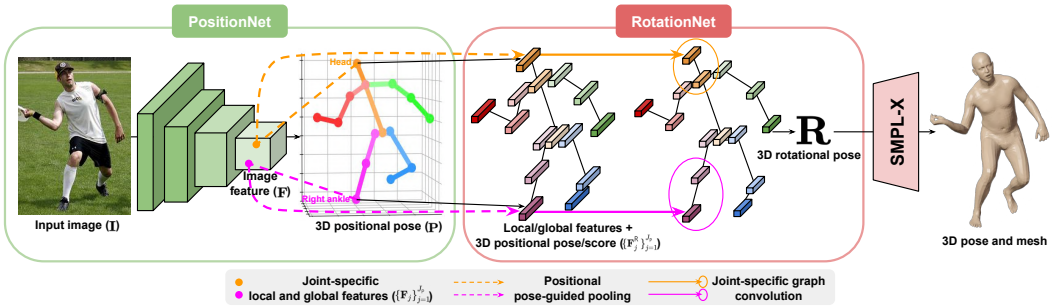
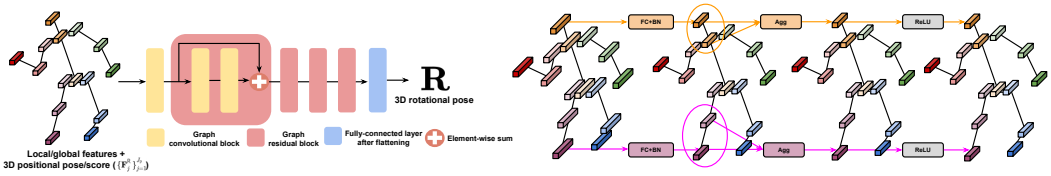


Figure 4.2: The overall pipeline of Pose2Pose, which consists of PositionNet and RotationNet. The PositionNet predicts the 3D positional pose. Then, the positional pose-guided pooling extracts the joint-specific local and global features. The RotationNet takes the joint-specific features with the 3D positional pose/scores and predicts the 3D rotational pose by the joint-specific graph convolution. The final 3D human pose and mesh are obtained by forwarding the predicted 3D human model parameters, including 3D rotational pose to a human model layer (*e.g.*, SMPL-X [6]). For the simplicity, we only illustrated body part Pose2Pose and head and right ankle operations.

convey depth information to the 3D rotational pose prediction module. On the other hand, our Pose2Pose utilize both joint-specific local and global features and 3D positional pose for the 3D rotational pose prediction. Especially, ours is for greatly challenging expressive 3D human pose and mesh estimation, while they are only for the body part.

Our Pose2Pose greatly improves the previous regression-based network by utilizing both local and global features for accurate expressive 3D human pose and mesh estimation. To this end, we combine the regression-based network with the detection-based network. Our detection-based network, PositionNet, provides the 3D positional pose. Then, we extract the local and global features on the predicted positional pose of the ResNet output image feature by the positional pose-guided pooling. Our regression-based network, RotationNet, accurately predicts the 3D rotational pose by the joint-specific graph convolution from the local and global features.



(a) Network architecture of RotationNet (b) Graph convolutional block of RotationNet

Figure 4.3: (a) The network architecture of the RotationNet. (b) The pipeline of the graph convolutional block, which processes graph features by the joint-specific graph convolution and aggregates the graph features using the adjacency matrix. FC, BN, and Agg denote a fully connected layer, 1D batch normalization, and graph feature aggregation using the adjacency matrix, respectively. We visualize detailed operations of only head and right ankle for the simplicity.

4.3 Pose2Pose

Figure 4.2 shows the overall pipeline of the proposed Pose2Pose. Pose2Pose consists of PositionNet and RotationNet, which will be described in the following subsections.

4.3.1 PositionNet

The PositionNet is designed as a fully convolutional network, which predicts 3D positional pose (*i.e.*, 3D positions of human joints) $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_{J_\mathrm{P}}]^T \in \mathbb{R}^{J_\mathrm{P} \times 3}$ from an input image $\mathbf{I}$. J_P denotes the number of joints representing the 3D positional pose. x- and y-axis of $\mathbf{P}$ are defined in image space, and z-axis of it is defined in root joint (*i.e.*, pelvis for the body and wrist for the hand)-relative depth space. For this, PositionNet extracts image feature $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ from the input image by ResNet [53], where C, H, and W denote the number of channels, height, and width. Then, a 1-by-1 convolution predicts 3D heatmaps of human joints $\mathbf{H} \in \mathbb{R}^{J_\mathrm{P} \times D \times H \times W}$, where D denotes the depth dimension size. To predict the 3D heatmaps from the 2D feature map $\mathbf{F}$, the 1-by-1 convolution first predicts a tensor of shape $\mathbb{R}^{J_\mathrm{P} D \times H \times W}$, and we reshape the tensor to the shape of $\mathbf{H}$ following Sun *et al.* [28]. The 3D positional pose $\mathbf{P}$ is calculated from $\mathbf{H}$ by the soft-argmax operation [28] in a differentiable way.

4.3.2 RotationNet

The RotationNet is designed as a graph convolutional network (GraphCNN), which predicts the 3D rotational pose (*i.e.*, 3D rotations of human joints) $\mathbf{R} \in \mathbb{R}^{J_{\mathrm{R}} \times 3}$, as illustrated in Figure 4.3. J_{R} denotes the number of joints representing the 3D rotational pose, which is often different from J_{P}. To this end, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathbf{A})$, where $\mathcal{V}$ and $\mathbf{A}$ are graph vertices and an adjacency matrix, respectively. The graph vertices represent human joints, where $|\mathcal{V}| = J_{\mathrm{P}}$. The adjacency matrix $\mathbf{A} \in \{0, 1\}^{J_{\mathrm{P}} \times J_{\mathrm{P}}}$ is constructed based on the human skeleton hierarchy in a pre-processing stage and fixed during the training and testing stage.

The initial feature of a jth graph vertex $\mathbf{F}_j^{\mathrm{R}} \in \mathbb{R}^{C+4}$ is a concatenation of a joint-specific local and global feature $\mathbf{F}_j \in \mathbb{R}^C$, the predicted 3D positional pose of jth joint $\mathbf{p}_j \in \mathbb{R}^3$, and 3D positional pose prediction confidence of jth joint $c_j \in \mathbb{R}$. The joint-specific features provide semantic information to the graph, computed by the positional pose-guided pooling. Moreover, the 3D positional pose provides geometric evidence, which conveys essential human articulation information. Finally, the prediction confidence can tell whether the joint-specific image feature and the 3D positional pose of each joint are reliable or not. The initial features of all graph vertices $\{\mathbf{F}_j^{\mathrm{R}}\}_{j=1}^{J_{\mathrm{P}}}$ are processed by the joint-specific graph convolution. We provide detailed descriptions of the positional pose-guided pooling and joint-specific graph convolution below.

Positional pose-guided pooling. The positional pose-guided pooling computes joint-specific local and global features $\{\mathbf{F}_j\}_{j=1}^{J_{\mathrm{P}}}$ using the predicted 3D positional pose $\mathbf{P}$. Since the coordinates of $\mathbf{p}_j$, (x_j, y_j) are not integers, we obtain the jth joint feature $\mathbf{F}_j$ at position $\mathbf{p}_j$ using bilinear interpolation on the image feature map $\mathbf{F}$.

The interpolated feature $\mathbf{F}_j$ is obtained from the exact position of the joint j; thus, it contains a local feature. However, the interpolated feature is not restricted to the local feature because the large size of the receptive field of ResNet output makes $\mathbf{F}_j$ contain information around the position of joint j, as well. Thus, $\mathbf{F}_j$ contains both joint-specific local and global features. The 3D positional pose prediction confidences $\{c_j\}_{j=1}^{J_\mathrm{P}}$ are also obtained by performing the positional pose-guided pooling on the estimated 3D heatmap $\mathbf{H}$.

Joint-specific graph convolution. Our joint-specific graph convolution uses separated learnable weights for each graph vertex. Specifically, we define learnable weight matrices $\{W_j \in \mathbb{R}^{C_\mathrm{out} \times C_\mathrm{in}}\}_{j=1}^{J_\mathrm{P}}$ for all joints of each graph convolution layer, where C_in and C_out denotes input and output channel dimensions, respectively. Then, the output graph feature of joint j is obtained by $\mathbf{F}_j^\mathrm{out} = \sigma_\mathrm{ReLU}(\sum_{i \in \hat{\mathcal{N}}_j} \tilde{a}_{ji} \sigma_\mathrm{BN}(W_i \mathbf{F}_i^\mathrm{in}))$, where $\mathbf{F}_i^\mathrm{in}$ is the input graph feature of joint i. σ_ReLU and σ_BN denotes ReLU activation function and 1D batch normalization [57], respectively. $\hat{\mathcal{N}}_j$ is defined as $\mathcal{N}_j \cup \{j\}$, where $\mathcal{N}_j$ denotes neighbors of a vertex j. $\tilde{a}_{ji}$ is is an entry of the normalized adjacency matrix $\tilde{\mathbf{A}}$ at (j, i), where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-\frac{1}{2}}$. $\mathbf{D}$ is a diagonal matrix of $\mathbf{A} + \mathbf{I}$. The RotationNet follows the network architecture of Liu *et al.* [92], which consists of one graph convolutional block and four graph residual blocks. Each block consists of joint-specific graph convolution, 1D batch normalization, and ReLU activation function. All the graph features have a channel dimension of 128, except for that of the input and output features.

At the last part of the RotationNet, we flatten the graph features into a vector and use a single fully connected layer to predict 3D rotational pose $\mathbf{R}$.

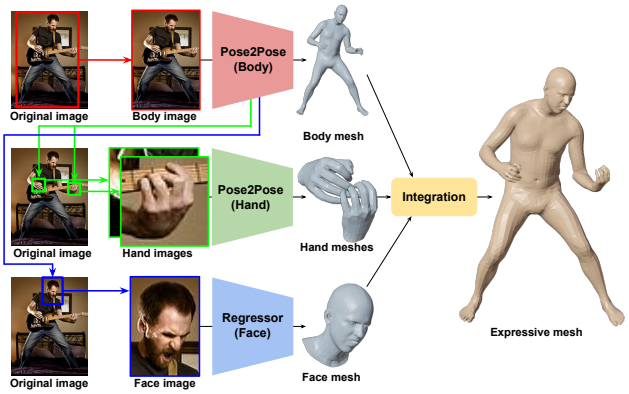


Figure 4.4: Our entire system for expressive 3D human pose and mesh estimation consists of three separated networks for the body, hand, and face. In the testing stage, the hand/face images are obtained using the predicted hand/face boxes from the body part. The integration module integrates the outputs of the three networks.

4.4 Expressive 3D human pose and mesh estimation

For the expressive 3D human pose and mesh estimation, we construct three separate networks for the body, hand, and face following Choutas *et al.* [7], as shown in Figure 4.4. This separation enables us to use part-specific datasets such as FreiHAND [4] and FFHQ [97]. Each network is responsible for each part, and an integration module integrates the outputs of each network in the testing stage. We provide descriptions of the networks of each part and the integration module below.

4.4.1 Body part

The body part uses Pose2Pose to predict 3D body global rotation $\theta_b^g \in \mathbb{R}^3$, 3D body rotational pose $\theta_b \in \mathbb{R}^{21\times3}$, shape parameter $\beta_b \in \mathbb{R}^{10}$, and camera parameter $k_b \in \mathbb{R}^3$. The shape parameter β_b represents human body shape identity (*e.g.*, thin/fat and short/tall), defined as coefficients of principal components in the human body shape space. θ_b^g and θ_b are predicted from RotationNet, and β_b and k_b are predicted from the global average pooled $\mathbf{F}$ using a separated fully connected layer. Instead

of directly predicting θ_b, we initially predict a latent code of VPoser [6] z_b and use VPoser to decode z_b to θ_b. The outputs are passed to the SMPL-X layer to obtain the final 3D body pose and mesh.

The body part additionally predicts hand and face bounding boxes to make the hand and face-cropped images during the testing stage. To this end, we concatenate the image feature $\mathbf{F}$ and 2D heatmap $\mathbf{H}'$ and pass it to two convolutional layers. The 2D heatmap $\mathbf{H}'$ is generated by making a Gaussian blob on the (x, y) position of $\mathbf{P}$. The soft-argmax [28] is applied to the output of the convolutional layers for the box centers. The widths and heights of the boxes are computed by performing positional pose-guided pooling on the box centers of $\mathbf{F}$ and pass the features of each box center to separated fully connected layers.

4.4.2 Hand part

We use exactly the same network architecture as that of the body part. The hand part outputs 3D hand global rotation $\theta_h^g \in \mathbb{R}^3$, 3D hand rotational pose $\theta_h \in \mathbb{R}^{15\times3}$, shape parameter $\beta_h \in \mathbb{R}^{10}$, and camera parameter $k_h \in \mathbb{R}^3$. The shape parameter β_h represents human hand shape identity (*e.g.*, thin/fat and small/large), defined as coefficients of principal components in the human hand shape space. The outputs are passed to the MANO layer to obtain the final 3D hand pose and mesh.

4.4.3 Face part

Unlike the joints of the body and hand, most of the face keypoints do not move according to 3D rotations of joints, making it hard to apply Pose2Pose. Instead, we design a simple regressor that consists of ResNet and fully connected layers. We perform GAP on the image feature $\mathbf{F}$ and fed it to separated fully connected

layers, which predict 3D face global rotation $\theta_f^g \in \mathbb{R}^3$, 3D jaw rotation $\theta_f \in \mathbb{R}^3$, shape parameter $\beta_f \in \mathbb{R}^{10}$, and expression code $\psi \in \mathbb{R}^{10}$. The shape parameter β_f represents human face shape identity (*e.g.*, thin/fat and small/large), defined as coefficients of principal components in the human face shape space. The expression code ψ represents human face expression, defined in a human expression latent space of Lin *et al.* [90]. The predicted parameters are passed to the FLAME layer to obtain the final 3D face pose and mesh.

4.4.4 Training the networks

The three networks of each part are trained separately. For all parts, we calculate $L1$ loss between predicted and groundtruth 3D positional pose following Moon and Lee [21]. In addition, $L1$ loss between predicted and groundtruth SMPL-X/MANO/FLAME parameters, 3D joint coordinates of SMPL-X/MANO/FLAME, and projected 2D joint coordinates are also calculated following Kolotouros [25]. For the hand and face box localization, we calculate $L1$ loss between predicted and groundtruth box centers, widths, and heights.

4.4.5 Integration of all parts in the testing stage

The final expressive 3D human pose and mesh is obtained by forwarding $\{\theta_b^g, \theta_b, \beta_b, \theta_{rh}^g, \theta_{rh}, \theta_{lh}^g, \theta_{lh}, \theta_f, \psi\}$ to SMPL-X, where $*_{rh}$ and $*_{lh}$ denote $*$ is from right and left hand, respectively. The 3D hand pose parameter θ_h of MANO and 3D jaw rotation θ_f and face expression code ψ of FLAME are compatible with those of SMPL-X; thus, we use them for the final prediction. As the body part often predicts wrong rotations of elbows and wrists in the roll axis, we selectively use the 3D hand global rotation θ_h^g to replace rotations of the elbows and wrists.

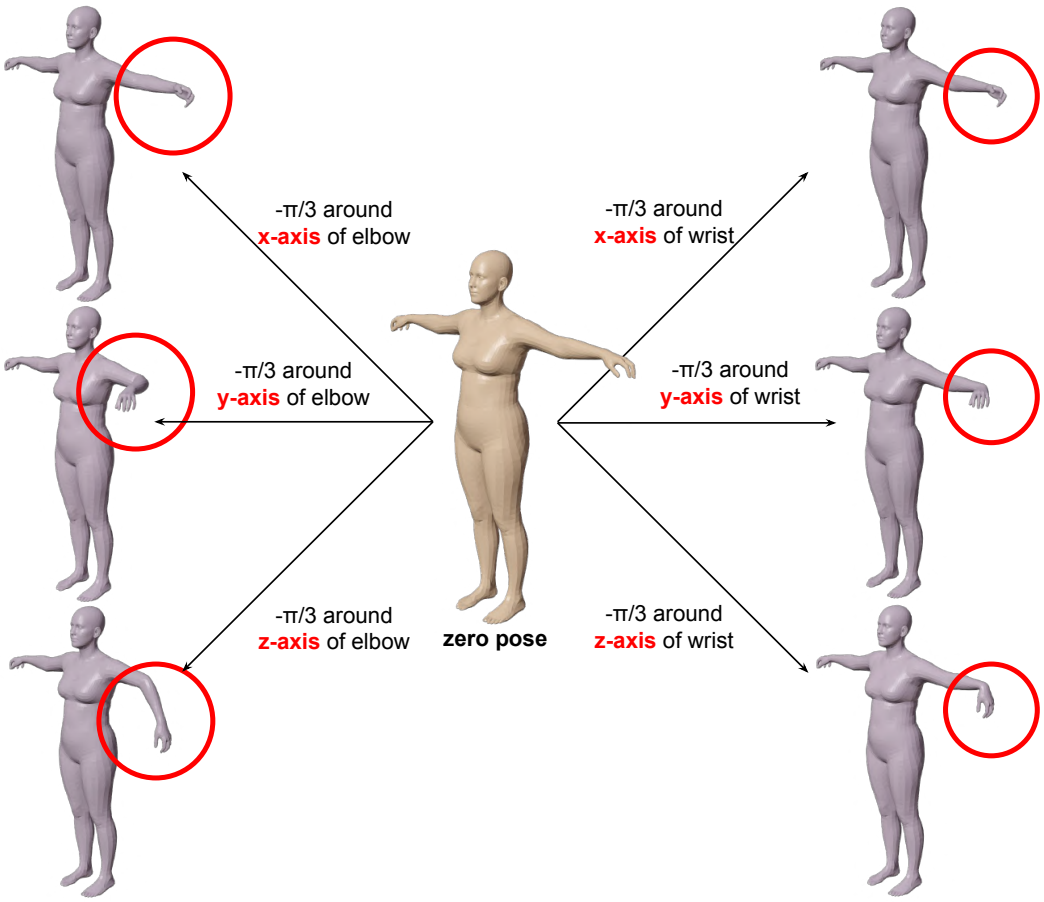


Figure 4.5: Visualized rotations of the elbow and wrist in each axis.

Algorithm 1 Integration of body and hands

Input: θ_b^g: Global rotation of body
Input: $\theta_b = \{\theta_j^l\}_{j=1}^{21}$: Local rotations of body joints
Input: $\theta_{rh}^g, \theta_{lh}^g$: Global rotations of right and left hands
Output: θ_b: Updated local rotations of body joints

1: Compute global rotations of body joints $\{\theta_j^g\}_{j=1}^{21}$ from θ_b^g and θ_b by forward kinematics
2: Let re, le, rw, lw denote joint index of right elbow, left elbow, right wrist, and left wrist, respectively.
3: **for** $(e, w, h) \leftarrow ((re, rw, rh), (le, lw, lh))$ **do**
4: $\quad\theta_w^g \leftarrow \theta_h^g$
5: $\quad x$-axis of $\theta_e^g \leftarrow x$-axis of $(\theta_e^g + \theta_h^g)/2$
6: $\quad$Compute new local rotations of elbow and wrist, $\hat{\theta}_e^l$ and $\hat{\theta}_w^l$, respectively, from $\{\theta_j^g\}_{j=1}^{21}$ by reversing forward kinematics
7: $\quad$**if** $|y$-axis of $\hat{\theta}_w^l| < \pi/4$ and $|z$-axis of $\hat{\theta}_w^l| < \pi/2$ **then**
8: $\quad\quad$Update $\theta_w^l \leftarrow \hat{\theta}_w^l$
9: $\quad\quad$Update $\theta_e^l \leftarrow \hat{\theta}_e^l$
10: $\quad$**end if**
11: **end for**

Algorithm 1 and Figure 4.5 show the integration procedure and how rotations of elbow and wrist change the body, respectively. First, we perform forward kinematics to compute global rotations of all body joints, including wrists and elbows (line 1). Then, we replace the global rotations of wrists and elbows using the global rotation of hands (lines 4 and 5). The replacement assumes x-axis rotations (roll of Euler angle) of the wrist and elbow are almost the same, which follows the anatomical structure of the human body, as shown in Figure 4.5. To avoid a sudden change of the elbow rotation, which can cause artifacts, we use an average rotation of the elbow and wrist (line 5). From the replaced global rotations of wrists and elbows, we compute new local rotations of wrists and elbows (line 6). Finally, we check the new local rotation follows the anatomical structure of the human body (line 7), where y- and z-axis rotations are shown in Figure 4.5. If true, we update the local

rotations of body joints, which become the final output of the integration (lines 8 and 9). We convert the 3D rotation of joints to Euler angles in line 3 - line 11. The integration is only performed when the distance between the center of the hand box and predicted wrist position is smaller than the box scale. If the distance is longer than the threshold, we consider the hand is not detected. In that case, we ignore all outputs from the hand part Pose2Pose by skipping Algorithm 1 and setting the hand pose to zero.

4.5 Implementation details

PyTorch [58] is used for implementation. The backbone part is initialized with the publicly released ResNet50 [53] pre-trained on ImageNet [59]. The weights are updated by Adam optimizer [60] with a mini-batch size of 192. The human body region is cropped using groundtruth box in both of training and testing stages following previous works [5, 23, 25]. The hand and face images are cropped from the original image using groundtruth box in the training stage and the predicted box in the testing stage. The cropped image is resized to 256×256. Data augmentations, including scaling, rotation, random horizontal flip, and color jittering, are performed in training. All the 3D rotations except for θ_b are initially predicted in the 6D rotational representation of Zhou *et al.* [98] and converted to the 3D axis-angle rotations. We flipped all hands to the right hand during the training and testing stage of the hand part. The initial learning rate is set to 10^{-4} and reduced by a factor of 10 at the 10^{th} epoch. We train each body, hand, and face part separately for 12 epochs with four NVIDIA RTX 2080 Ti GPUs.

4.6 Experiment

4.6.1 Training sets and evaluation metrics

Training sets. To train the body part, we use Human3.6M [2], MPI-INF-3DHP [14], MSCOCO [1,99], and MPII [63]. For the hand part, FreiHAND [4], InterHand2.6M [9], and MSCOCO [99] are used for the training. Finally, FFHQ [97] and MSCOCO [99] are used for the face part training. We obtained SMPL/SMPL-X/MANO/FLAME fits of the datasets using SMPLify-X [6] and used them as pseudo-groundtruths. We will release all fits for the reproducible and continual study.

Evaluation metrics. MPJPE and MPVPE are widely used to evaluate 3D human body/hand pose and mesh estimation, where each calculates the average 3D joint distance (mm) and 3D mesh vertex distance (mm) between predicted and groundtruth, respectively, after aligning a root joint translation. PA MPJPE and PA MPVPE further align a rotation and scale. F-score is additionally used for the 3D hand pose and mesh estimation evaluation. For the face part, the average of the closest distance between a predicted 3D face mesh vertex and groundtruth 3D face scan point is used.

4.6.2 Ablation study

For the ablation study, we train the body part on Human3.6M, MSCOCO, and MPII and report errors on 3DPW, which is a standard experimental protocol of recent 3D human body pose and mesh estimation works. We use SMPL for the human model of the body part. For the hand part ablation study, we train the hand part Pose2Pose on FreiHAND, MSCOCO, and InterHand2.6M and report errors on the FreiHAND validation set. For the expressive whole-body ablation study, we test

How to pool?	How to process?	PA MPJPE
GAP	FC	60.7 [23–25]
	GraphCNN	59.5
GAP+PPP	FC	57.4
	GraphCNN	57.6
PPP	FC	57.5
	GraphCNN	**56.8 (Ours)**

Table 4.1: PA MPJPE comparison between models with various pooling methods and processing modules on 3DPW.

How to pool?	How to process?	PA MPJPE	PA MPVPE
GAP	FC	6.7	6.5
PPP	GraphCNN	**5.4**	**5.2**

Table 4.2: PA MPJPE and PA MPVPE comparison between the previous widely used approach (first row) [4, 15–17] and our approach (second row) on FreiHAND.

our entire system on EHF.

Effectiveness of the positional pose-guided pooling. We show the effectiveness of the positional pose-guided pooling (PPP) in Table 4.1. For this, we report PA MPJPE of our Pose2Pose and its variants that use GAP or a combination of GAP and PPP. As the table shows, adding PPP to GAP or replacing GAP with PPP decrease the error regardless of the processing modules, fully connected layer (FC) and GraphCNN. It is noticeable that our PPP achieves significantly lower error compared with the combination of GAP and FC, the most widely used one in previous works [23–25]. Interestingly, replacing GAP with PPP achieves a better result than adding PPP to GAP when the GraphCNN is used. This is because a global image feature from the GAP contains much unnecessary information, such as backgrounds, which makes the performance worse. On the other hand, the local and global features from the PPP contains essential human articulation information, and the GraphCNN aggregates the features by considering the human skeleton hierarchy, which makes the aggregated feature highly useful. Table 4.2 shows the

How to pool?	Which graph conv.?	PA MPJPE
GAP	Shared graph conv.	61.5 [5]
	Joint-specific graph conv.	59.5
PPP	Shared graph conv.	64.5
	Joint-specific graph conv.	**56.8 (Ours)**

Table 4.3: PA MPJPE comparison between models with various pooling methods and graph convolutions on 3DPW.

Image feature	Joint coordinate + confidence	PA MPJPE
✓	✗	58.0
✗	✓	59.2
✓	✓	**56.8 (Ours)**

Table 4.4: PA MPJPE comparison between models with various input combinations of the RotationNet on 3DPW.

same tendency on the hand part. The comparisons clearly show the benefit of PPP, which preserves joint-specific local and global features, while GAP cannot. For the experiment, we used the joint-specific graph convolution for all GraphCNN. When PPP is used, the local and global image features of joints and joint coordinates with the confidence are used for the final 3D rotational pose prediction. When both GAP and PPP are used, the FC takes a concatenation of flattened $\{\mathbf{F}_j^{\mathrm{R}}\}_{j=1}^{J_{\mathrm{P}}}$ and the output vector of GAP. On the other hand, the GraphCNN takes a graph, where jth node is a concatenation of $\mathbf{F}_j^{\mathrm{R}}$ and the output vector of GAP.

Effectiveness of the joint-specific graph convolution. Table 4.3 shows the benefit of the joint-specific graph convolution. To this end, we report PA MPJPE of Pose2Pose and its variants that use GAP or the shared graph convolution. The shared graph convolution uses shared learnable weights for all graph vertices like the vanilla graph convolution [91]. Our joint-specific graph convolution achieves a lower error than the shared graph convolution regardless of the pooling method, GAP and PPP. Especially, the combination of the PPP and the joint-specific graph convolution significantly outperforms a combination of GAP and shared graph convolution, used

Figure 4.6: Qualitative results of our framework on MSCOCO validation set.

in Kolotouros *et al.* [5]. We also found that the shared graph convolution increases the error a lot when the input features are from PPP. This is because the shared graph convolution applies the same weights to features of all graph vertices, while each feature of a graph vertex from PPP has distinctive joint-specific information. The comparisons clearly show the benefit of the joint-specific graph convolution.

Inputs of the RotationNet. We show how each input of the RotationNet affects the accuracy in Table 4.4. The table shows that taking both the image feature and joint coordinate with confidence achieves the best accuracy. The image feature extracted by PPP can provide local/global contextual information, and the joint coordinate with confidence predicted by PositionNet can provide 3D geometric information. We design our RotationNet to take both inputs, thus can utilize both local/global contextual information and 3D geometric information. The comparison clearly shows the validity of our RotationNet design.

What PositionNet predicts?	PA MPJPE
2D positional pose	58.2
3D positional pose	**56.8 (Ours)**

Table 4.5: PA MPJPE comparison between models with various output of the PositionNet on 3DPW.

Integration method	PA MPVPE
Without prior	55.5
With prior	**51.9 (Ours)**

Table 4.6: PA MPVPE comparison between models without and with the anatomical prior during the integration on EHF.

Effectiveness of the 3D positional pose. We show the effectiveness of utilizing the 3D positional pose over the 2D positional pose in Table 4.5. The table shows that PA MPJPE gets better when PositionNet predicts 3D positional pose than 2D positional pose. This is because additional depth information is provided to the RotationNet, which helps to resolve the depth ambiguity. The comparison clearly shows the effectiveness of predicting 3D positional pose from the PositionNet.

Effectiveness of the anatomical prior during the integration. We show the effectiveness of the anatomical prior during the integration, described in line 7 of Algorithm 1 in Table 4.6. The table shows that our anatomical prior significantly reduces PA MPVPE. The comparison clearly shows the benefit of the anatomical prior.

4.6.3 Comparison with state-of-the-art methods

Body part. Table 4.7 shows comparison between our body part Pose2Pose and previous state-of-the-art methods on 3DPW [80]. It shows our Pose2Pose significantly outperforms previous works by a large margin, including both body-only methods and the expressive method [7]. Following previous works [21,22,25], we use SMPL for the human model, and 14 joints are used for the evaluation. In addition, we report

Methods	Scale	MPJPE	PA MPJPE
HMR [23]	Body only	130.0	81.3
HMMR [85]		-	72.6
GraphCMR [5]		-	70.2
Arnab *et al.* [86]		-	72.2
SPIN [25]		96.9	59.2
Pose2Mesh [22]		**88.9**	58.3
I2L-MeshNet [21]		93.2	57.7
ExPose [7]	All parts	93.4	60.7
Pose2Pose (Ours)		89.4	**55.5**
Pose2Pose* (Ours)		**84.8**	**52.9**

Table 4.7: MPJPE and PA MPJPE comparison on 3DPW. * denotes its ResNet is initialized with that of SimpleBaseline [18].

Methods	Scale	PA errors	F scores
Hasson *et al.* [16]	Hand only	13.2 / -	0.436 / 0.908
Boukhayma *et al.* [15]		13.0 / -	0.435 / 0.898
FreiHAND [4]		10.7 / -	0.529 / 0.935
Pose2Mesh [22]		7.8 / 7.7	0.674 / 0.969
I2L-MeshNet [21]		7.6 / **7.4**	0.681 / 0.973
ExPose [7]	All parts	11.8 / 12.2	0.484 / 0.918
Pose2Pose (Ours)		**7.4** / **7.4**	**0.683** / **0.974**

Table 4.8: PA MPVPE/PA MPJPE and F-score@5mm/15mm comparison on FreiHAND.

the performance of another Pose2Pose, of which ResNet part is initialized with the pre-trained weights of 2D human pose estimation network [18] on MSCOCO. The table shows that initializing the ResNet part, included in PositionNet, with a pre-trained 2D human pose estimation network significantly boosts the performance. We think this is because the pre-trained 2D human pose estimation network already can provide accurate 2D pose; thus, it can converge to a better 3D positional pose estimation network. This shows better PositionNet can lead to significant performance gain in our framework.

Hand part. Table 4.8 shows a comparison between our hand part Pose2Pose and previous state-of-the-art methods on FreiHAND [4]. It shows our Pose2Pose achieves

Methods	Scale	Mean	Median	Std.
RingNet [100]	Face only	2.08/2.02	1.63/1.58	1.79/**1.68**
ExPose [7]	All parts	2.27/2.42	1.76/1.91	1.97/2.03
Regressor (Ours)		**2.02/1.99**	**1.55/1.53**	**1.78**/1.76

Table 4.9: Mean, median, and standard deviation of 3D face mesh error comparison on low-quality/high-quality images of Stirling.

Methods	PA MPVPE			PA MPJPE	
	All	Hands	Face	Body	Hands
SMPLify-X [6]	65.3	12.3	6.3	87.6	12.9
MTC [93]	67.2	-	-	107.8	16.7
ExPose [7]	54.5	12.8	5.8	62.8	13.1
Pose2Pose (Ours)	**51.9**	**12.0**	**5.6**	**62.6**	**11.8**

Table 4.10: PA MPVPE and PA MPJPE comparison on EHF. The numbers in hands are averaged values of left and right hands.

comparable accuracy with a recent state-of-the-art hand-only method [21] and significantly outperforms the expressive method [7].

Face part. Table 4.9 shows comparison between our face part regressor and previous state-of-the-art methods on Stirling [101]. It shows our regressor achieves lower errors compared with the face-only method and expressive method [7].

All parts. Table 4.10 shows comparison between our Pose2Pose and previous expressive methods on EHF [6]. SMPL-X is used for the human model of the body part. For the evaluation, we integrated body, hand, and face parameters by our integration module, described in Section 4.4.5. The table shows our Pose2Pose outperforms previous methods by a large margin. Figure 4.8 shows qualitative comparison with previous state-of-the-art expressive method, ExPose [7], with ours on MSCOCO validation set. Pose2Pose recovers much more accurate expressive 3D pose and shape, including hands and face.

Taken together, our Pose2Pose outperforms all methods on all part-specific and expressive datasets. The comparisons clearly show the effectiveness of Pose2Pose,

Figure 4.7: Qualitative results on internet images. From top to bottom, left to right, the persons in the images are Freddie Mercury of band Queen, Lady Gaga, Adele, Dave Mustaine of band Megadeth, James Hetfield of band Metallica, David Draiman of band Disturbed, Lisa Su of AMD, Jensen Huang of NVIDIA, Steven Ogg of GTA 5, Steven Jobs of Apple, Elon Musk of Tesla, and Mark Zuckerberg of Facebook.

Figure 4.8: Qualitative comparison with ExPose [7] on MSCOCO validation set. Pose2Pose recovers much more accurate expressive 3D pose and shape, including hands and face.

which benefit from our novel positional pose-guided pooling and joint-specific graph convolution. Figure 4.6 and 4.7 show qualitative results of Pose2Pose on MSCOCO and internet images, respectively. Figure 4.9 shows expressive 3D multi-person pose and mesh estimation result on in-the-wild images of MSCOCO. The framework of Moon *et al.* [26] is used to extend the expressive single person 3D pose and mesh to the multi-person case.

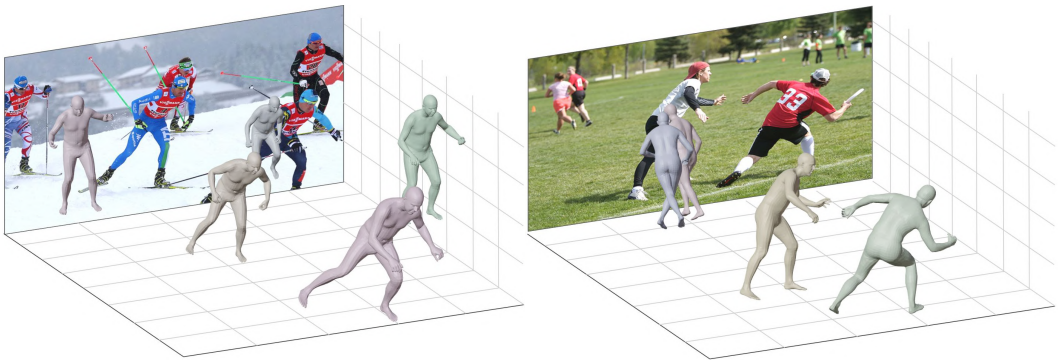


Figure 4.9: Expressive 3D multi-person pose and mesh estimation result on in-the-wild images.

4.6.4 Running time

From a single RGB image, the body and hand part Pose2Pose take 0.06 and 0.07 seconds per frame, respectively, and the face regressor takes 0.02 seconds per frame. The hand part takes the left and right hand images simultaneously. The integration module takes 0.01 seconds per frame, which includes the forwarding time to the SMPL-X layer. In total, our whole framework runs at 6.3 frames per second for expressive 3D human pose and mesh estimation from a single RGB image. This is the same running time as that of the previous expressive method, ExPose [7]. The running times are measured by using a single RTX 2080 Ti GPU and making the mini-batch size 1.

4.7 Conclusion

We present Pose2Pose, a 3D positional pose-guided 3D rotational pose prediction network for expressive 3D human pose and mesh estimation from a single RGB image. In contrast to previous works that rely on only a global image feature, ours utilize joint-specific local and global features, extracted by the positional pose-guided pooling, with joint-specific graph convolution. We apply our Pose2Pose for expressive

3D human pose and mesh estimation and achieved state-of-the-art accuracy on all part-specific and expressive datasets.

Chapter 5

Conclusion and Future Work

5.1 Summary and Contributions of the Dissertation

In this dissertation, three kinds of novel approaches towards expressive 3D multi-person pose and shape estimation from a single image were introduced, which are based on 1) 3D multi-person body pose estimation, 2) 3D multi-person pose and shape estimation, and 4) 3D pose and shape estimation for the integrated body, hands, and face.

In Chapter 2, a camera distance-aware 3D multi-person body pose estimation framework [26] was proposed. The proposed RootNet [26] computes a relative position between the camera and all persons by refining a pre-defined human scale (2 meters $\times$ 2 meters) using a deep image feature. As the deep image feature contains both pose and appearance information, it can refine the pre-defined scale, which can differ by pose and appearance variations of humans. The proposed whole framework [26], which consists of RootNet and state-of-the-art human detection and 3D single person body pose estimation methods, achieves state-of-the-art performance

on various datasets. The framework is also used for the following approaches in this dissertation to extend them to the multi-person case.

In Chapter 3, I proposed I2L-MeshNet [21], which predicts heatmaps of mesh vertices instead of 3D rotations of human joints for accurate 3D human pose and shape estimation. To reduce drastic GPU usage that arises from predicting heatmaps for all mesh vertices, the lixel-based 1D heatmap is used as a prediction target instead of the voxel-based 3D heatmap. Experimental results demonstrate the effectiveness of the lixel-based 1D heatmap compared with 3D rotations of human joints and voxel-based 3D heatmaps.

Finally, in Chapter 4, I proposed a framework [20] for expressive 3D human pose and shape estimation. Although the above described I2L-MeshNet achieves highly accurate performance, 3D rotations of human joints are needed for many computer graphical applications, such as animations. To this end, Pose2Pose [20], 3D positional pose-guided 3D rotational pose prediction network, is designed for accurate 3D rotational pose prediction. The outputs of body, hand, and face part networks are integrated for the expressive 3D human pose and shape. The experimental results show that Pose2Pose achieves state-of-the-art performance on all part-specific and expressive datasets.

5.2 Future Directions

Although the above methods are demonstrated to be highly effective, there is room for improvement.

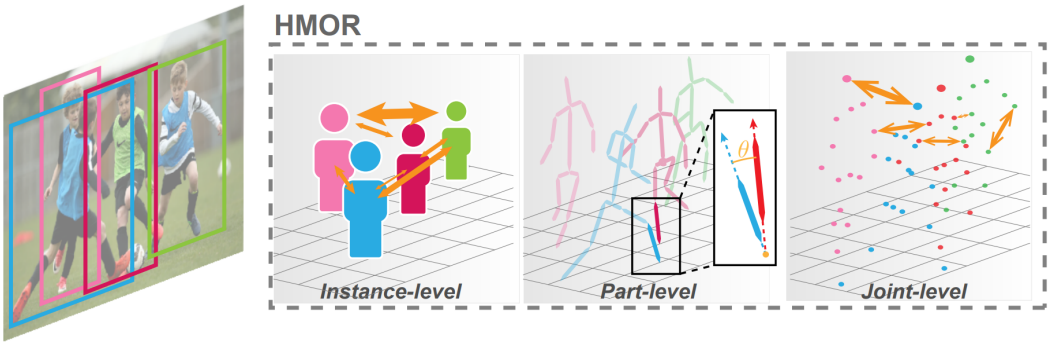


Figure 5.1: Global context-aware 3D multi-person pose estimation of HMOR [8].

5.2.1 Global Context-Aware 3D Multi-Person Pose Estimation

To compute the relative position between the camera and persons, RootNet takes a cropped single person image, thus ignoring global contextual information outside of the human area. The global information can contain other persons, objects, or background scenes, which can be helpful in determining the absolute depth value of the human. For example, if a person is occluded by objects, we can guess that the objects are closer to the camera than the person. Utilizing such information in a weakly-supervised way [8] can be a promising future work, as shown in Figure 5.1.

5.2.2 Unified Framework for Expressive 3D Human Pose and Shape Estimation

The proposed framework for expressive 3D human pose and shape estimation consists of three separate networks, which takes body, hands, and face images, respectively, as shown in Figure 5.2. To make it easier to use and lighter, the three networks should be unified into a single one, which also enables the network to utilize image features outside of the hand/face area when predicting hand/face 3D poses. The image feature outside the area will be especially helpful for the hand part when a hand is occluded or suffers from severe motion blur because the network can guess

a probable hand pose from the outside image feature. Unifying can be achieved by adding hand and face branches to the body branch. However, unifying can degrade the hand and face part accuracy because input hand and face image resolutions will be largely decreased compared with those of the proposed framework that employs separated networks and takes cropped and resized hand and face images. In addition, using part-specific datasets [2, 9] becomes difficult because a single network should handle all parts. Nevertheless, I think integrating the separated three networks into a single system is a promising future work.

5.2.3 Enhancing Appearance Diversity of Images Captured from Multi-View Studio

Images captured from multi-view studio are paired with GT 3D poses; however they have monotonous appearances [2, 9], which are far from those of in-the-wild images, as shown in Figure 5.3. Thus, a model trained on the images often fails to generalize to in-the-wild images. Although making a mini-batch with half from multi-view studio datasets and a half from in-the-wild datasets [1] can resolve this issue to some degree, there is still a large image appearance domain gap. Recent advancement of the pose transfer [102] and image translation [103] can be used to enhance the appearance diversity of the images of multi-view datasets to those of in-the-wild images. In particular, as hand images often suffer from many image degradation issues, adding image degradation, such as motion blur, is necessary for 3D hand pose and shape estimation. Recently, Moon *et al.* [104] proposed a framework to obtain pseudo-GT 3D human pose and shape from in-the-wild images, and this future work can complement their approach by enhancing the images captured from multi-view studio. Therefore, a model can enjoy both image appearance diversity

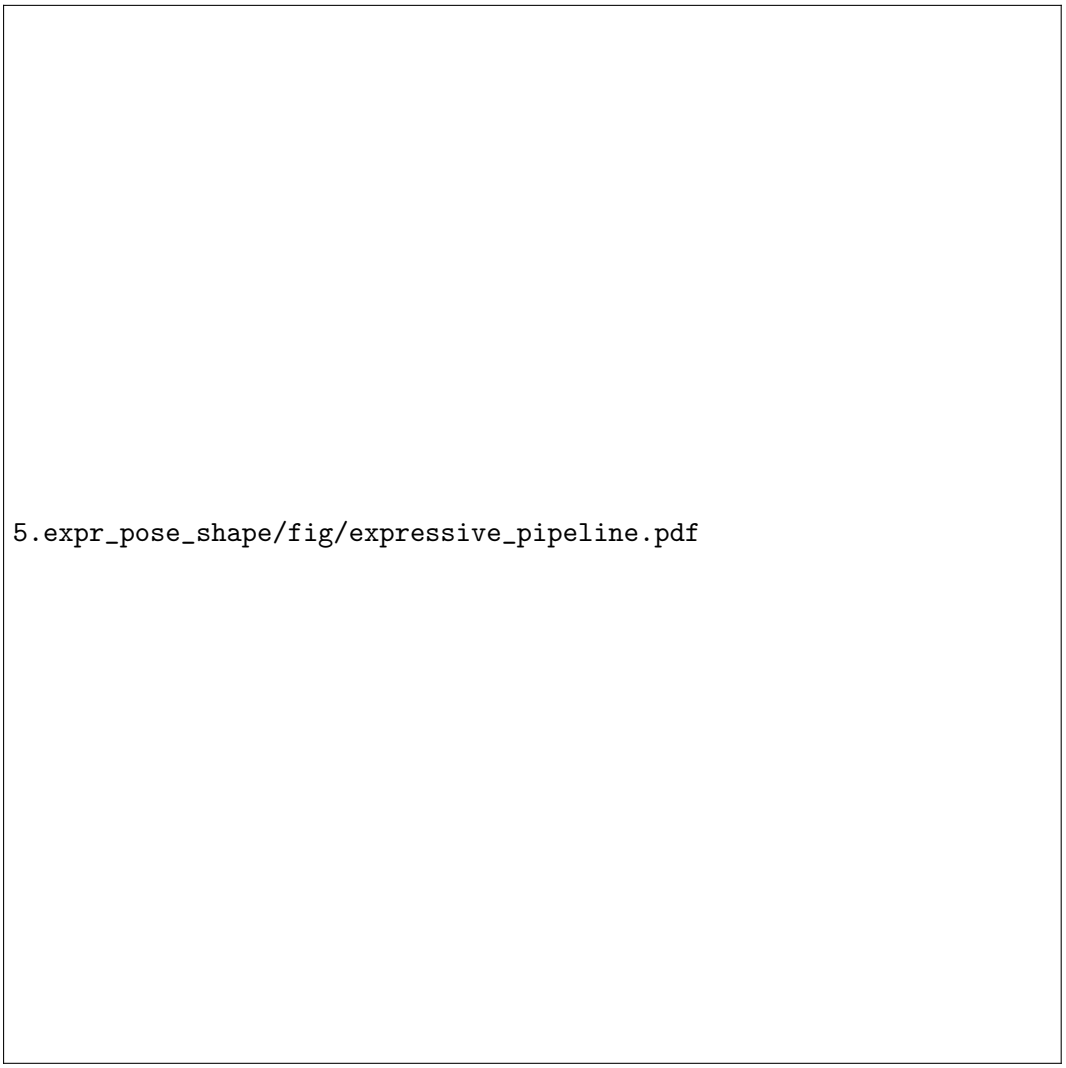


Figure 5.2: Expressive 3D human pose and shape estimation pipeline of Chapter 4 consisting of three separated networks.

Figure 5.3: Appearance comparison between images from (a) multi-view datasets [2, 9] and (b) in-the-wild datasets [1].

and strong 3D supervisions.

5.2.4 Extension to the video for temporally consistent estimation

All the introduced approaches are single image-based ones. Although the results on video, obtained by applying them on each frame of the video, are reasonable, there are noticeable jitterings. Recent advancement of video-based 3D human pose and shape estimation [10] can be applied for the temporal consistency, as shown in Figure 5.4.

5.2.5 3D clothed human shape estimation in the wild.

Current 3D human shape estimation methods have two separate directions: 1) 3D naked body shape estimation from images with diverse appearance and poses [20,21], 2) 3D clothed body shape estimation from images with simple appearance and poses [11]. Figure 5.5 shows 3D clothed human shape reconstruction results of PIFuHD [11]. The reason for this separated direction is that obtaining GT 3D naked

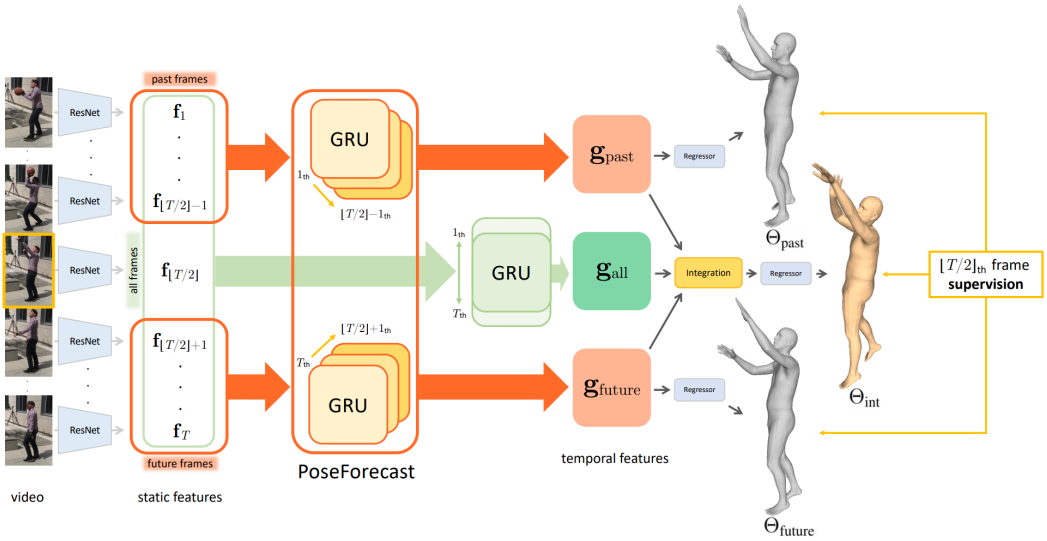


Figure 5.4: Temporally consistent 3D human pose and shape estimation network of TCMR [10].

Figure 5.5: Reconstructed 3D clothed human shapes of PIFuHD [11].

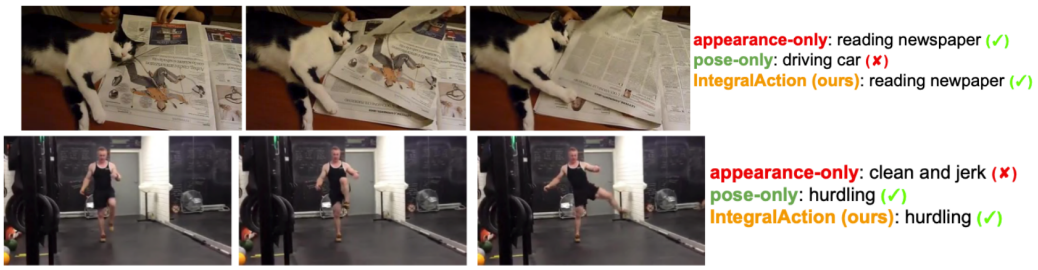


Figure 5.6: Robust action recognition of IntegralAction [12] on both in-context and out-of-context action videos.

body shape from in-the-wild images is possible to some extent by fitting parametric body models to the GT 2D poses, however obtaining GT 3D clothed body shape is only possible when 3D scan data is provided, which is not applicable to the in-the-wild environment. The proposed approaches in this dissertation are focused on the first direction; however, I am planning to merge the second direction by designing a weakly-supervised setting and loss functions that can be applied on images without 3D scans. This new direction would be helpful for many applications, such as making a clothed personal avatar.

5.2.6 Robust human action recognition from a video.

The estimated 3D human pose and shape can be useful for human action recognition from a video. In particular, Weinzaepfel *et al.* [105] and Moon *et al.* [12] showed that the 3D pose and shape are especially useful when the input video contains out-of-context actions, as shown in Figure 5.6. The out-of-context action means a sequence of human motion, where the motion does not match the context of a video (*e.g.*, mime). As 3D pose and shape provide only geometric information without appearance information, a model that takes only 3D human pose and shape is not easily fooled by out-of-context action videos. However, as human actions can be

determined by objects or backgrounds, the lack of appearance information can make the input 3D pose and shape video ambiguous. On the other hand, a model that takes only an RGB video can be easily fooled by out-of-context action videos; for example, it predicts the human action class as *"swimming"* although a human is just standing in the swimming pool, while does not suffer from the context ambiguity. Recently, an action recognition system that takes RGB and 2D pose videos has been proposed for robust action recognition on both in-context and out-of-context action videos [12]. Although they successfully showed the robustness on both in-context and out-of-context action videos, the geometric information delivered by 2D pose video is not sufficient. It lacks depth information and only contains sparse body keypoint coordinates. Providing additional depth information with hand poses and facial expressions using the proposed approach can be a promising future work.

Bibliography

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

[2] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *TPAMI*, 2014.

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

[4] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "FreiHand: A dataset for markerless capture of hand pose and shape from single RGB images," in *ICCV*, 2019.

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

[6] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *CVPR*, 2019.

[7] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *ECCV*, 2020.

[8] J. Li, C. Wang, W. Liu, C. Qian, and C. Lu, "HMOR: Hierarchical multi-person ordinal relations for monocular multi-person 3D pose estimation," in *ECCV*, 2020.

[9] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image," in *ECCV*, 2020.

[10] H. Choi, G. Moon, and K. M. Lee, "Beyond static features for temporally consistent 3d human pose and shape from a video," *arXiv e-prints*, pp. arXiv–2011, 2020.

[11] S. Saito, T. Simon, J. Saragih, and H. Joo, "PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization."

[12] G. Moon, H. Kwon, K. M. Lee, and M. Cho, "Integralaction: Pose-driven feature integration for robust human action recognition in videos," *arXiv preprint arXiv:2007.06317*, 2020.

[13] G. Rogez, P. Weinzaepfel, and C. Schmid, "LCR-Net: Localization-classification-regression for human pose," in *CVPR*, 2017.

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

[15] A. Boukhayma, R. de Bem, and P. H. Torr, "3D hand shape and pose from images in the wild," in *CVPR*, 2019.

[16] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019.

[17] Y. Rong, T. Shiratori, and H. Joo, "FrankMocap: Fast monocular 3D hand and body motion capture by regression and integration," *arXiv preprint arXiv:2008.08324*, 2020.

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

[19] H. Joo, T. Simon, and Y. Sheikh, "Total Capture: A 3D deformation model for tracking faces, hands, and bodies," in *CVPR*, 2018.

[20] G. Moon and K. M. Lee, "Pose2Pose: 3D positional pose-guided 3D rotational pose prediction for expressive 3d human pose and mesh estimation," *arXiv preprint arXiv:2011.11534*, 2020.

[21] ——, "I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image," in *ECCV*, 2020.

[22] H. Choi, G. Moon, and K. M. Lee, "Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose," in *ECCV*, 2020.

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

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

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

[26] G. Moon, Y. C. Ju, and K. M. Lee, "Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image," in *ICCV*, 2019.

[27] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "XNect: Real-time multi-person 3D motion capture with a single RGB camera," *ACM TOG*, 2020.

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

[29] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *CVPR*, 2017.

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

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

[32] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Weaklysupervised transfer for 3d human pose estimation in the wild," in *ICCV*, 2017.

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

[34] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image," in *ICCV*, 2019.

[35] G. Moon, Y. C. Ju, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *CVPR*, 2018.

[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[37] S. Huang, M. Gong, and D. Tao, "A coarse-fine network for keypoint localization," in *ICCV*, 2017.

[38] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *CVPR*, 2018.

[39] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *CVPR*, 2017.

[40] G. Moon, J. Y. Chang, and K. M. Lee, "PoseFix: Model-agnostic general human pose refinement network," in *CVPR*, 2019.

[41] ——, "Multi-scale aggregation r-cnn for 2D multi-person pose estimation," in *CVPRW*, 2019.

[42] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *CVPR*, 2016.

[43] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *ECCV*, 2016.

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

[45] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *NeurIPS*, 2017.

[46] M. Kocabas, S. Karagoz, and E. Akbas, "MultiPoseNet: Fast multi-person pose estimation using pose residual network," in *ECCV*, 2018.

[47] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," in *ACCV*, 2014.

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

[49] S. Park, J. Hwang, and N. Kwak, "3D human pose estimation using convolutional neural networks with 2D pose information," in *ECCV*, 2016.

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

[51] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3D pose estimation," in *AAAI*, 2018.

[52] J. Y. Chang and K. M. Lee, "2D-3D pose consistency-based conditional random fields for 3D human pose estimation," *CVIU*, 2018.

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

[54] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[55] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, "Detectron," https://github.com/facebookresearch/detectron, 2018.

[56] F. Massa and R. Girshick, "maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch," https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

[57] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[58] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2014.

[61] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[62] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, 1975.

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

[64] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[65] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, "A dual-source approach for 3D pose estimation from a single image," in *CVPR*, 2016.

[66] F. Moreno-Noguer, "3D human pose estimation from a single image via distance matrix regression," in *CVPR*, 2017.

[67] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Monocap: Monocular human motion capture using a CNN coupled with a geometric prior," *TPAMI*, 2019.

[68] G. Rogez, P. Weinzaepfel, and C. Schmid, "LCR-Net++: Multi-person 2D and 3D pose detection in natural images," *TPAMI*, 2019.

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

[70] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[71] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM TOG*, 2015.

[72] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM TOG*, 2017.

[73] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3D hand shape and pose estimation from a single RGB image," in *CVPR*, 2019.

[74] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NeurIPS*, 2014.

[75] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *ECCV*, 2016.

[76] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural Body Fitting: Unifying deep learning and model based human pose and shape estimation," in *3DV*, 2018.

[77] Y. Xu, S.-C. Zhu, and T. Tung, "DenseRaC: Joint 3D pose and shape estimation by dense render-and-compare," in *ICCV*, 2019.

[78] G. Pavlakos, N. Kolotouros, and K. Daniilidis, "TexturePose: Supervising human mesh estimation with texture consistency," in *ICCV*, 2019.

[79] S. Baek, K. In Kim, and T.-K. Kim, "Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering," in *CVPR*, 2019.

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

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

[82] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *ECCV*, 2018.

[83] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz, "Hand pose estimation via latent 2.5D heatmap regression," in *ECCV*, 2018.

[84] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3D and 2D human representations," in *CVPR*, 2017.

[85] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3D human dynamics from video," in *CVPR*, 2019.

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

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

[88] ——, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*, 2011.

[89] H. Joo, N. Neverova, and A. Vedaldi, "Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation," *arXiv preprint arXiv:2004.03686*, 2020.

[90] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans." *ACM TOG*, 2017.

[91] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[92] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang, "A comprehensive study of weight sharing in graph networks for 3D human pose estimation," in *ECCV*, 2020.

[93] D. Xiang, H. Joo, and Y. Sheikh, "Monocular Total Capture: Posing face, body, and hands in the wild," in *CVPR*, 2019.

[94] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "GHUM & GHUML: Generative 3D human shape and articulated pose models," in *CVPR*, 2020.

[95] R. A. Guler and I. Kokkinos, "HoloPose: Holistic 3D human reconstruction in-the-wild," in *CVPR*, 2019.

[96] H. Zhang, J. Cao, G. Lu, W. Ouyang, and Z. Sun, "DaNet: Decompose-and-aggregate network for 3D human shape and pose estimation," in *ACM MM*, 2019.

[97] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019.

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

[99] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, "Whole-body human pose estimation in the wild," in *ECCV*, 2020.

[100] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *CVPR*, 2019.

[101] Z.-H. Feng, P. Huber, J. Kittler, P. Hancock, X.-J. Wu, Q. Zhao, P. Koppen, and M. Rätsch, "Evaluation of dense 3D reconstruction from 2D face images in the wild," *FG*, 2018.

[102] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *CVPR*, 2019.

[103] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[104] G. Moon and K. M. Lee, "Neuralannot: Neural annotator for in-the-wild expressive 3d human pose and mesh training sets," *arXiv preprint arXiv:2011.11232*, 2020.

[105] P. Weinzaepfel and G. Rogez, "Mimetics: Towards understanding human actions out of context," *arXiv preprint arXiv:1912.07249*, 2019.

국문초록

인간은 우리의 일상생활에서 가장 중심이 되고 흥미로운 대상이다. 그에 따라 모션 캡처, 인간-컴퓨터 인터렉션 등 많은 인간중심의 기술과 학문이 산업계와 학계에서 제안되었다. 인간의 정확한 3D 기하 (즉, 인간의 3D 자세와 형태)를 복원하는 것은 인간중심 기술과 학문에서 가장 중요한 부분 중 하나이다. 카메라의 빠른 대중화로 인해 단일 이미지는 많은 알고리즘의 널리 쓰이는 입력이 되었고, 그로 인해 많은 단일 이미지 기반의 3D 인간 자세 및 형태 추정 알고리즘이 제안되었다.

손과 발을 포함한 전신의 3D 자세와 형태는 인간의 의도와 느낌을 포함한 표현적이고 풍부한 정보를 제공한다. 하지만 전신의 3D 자세와 형태를 복원하는 것은 매우 어렵기 때문에 오직 극소수의 방법만이 이를 풀기 위해 제안되었고, 이를 위한 방법들을 표현적인 방법이라고 부른다. 표현적인 3D 자세와 형태를 한 번에 복원하는 것 대신, 사람의 몸, 손, 그리고 얼굴의 3D 자세와 형태를 따로 복원하는 방법들이 제안되었다. 이러한 방법들을 부분 특유 방법이라고 부른다. 이러한 문제의 간단화 이외에도 몇 가지의 간단화가 더 존재한다. 예를 들어, 많은 방법은 3D 형태를 제외한 3D 자세만을 추정한다. 이는 추가적인 3D 형태 추정이 문제를 더 어렵게 만들기 때문이다. 또한, 대부분의 방법은 오직 단일 사람의 경우만 고려하고 여러 사람의 경우는 고려하지 않는다. 그러므로, 현재 제안된 방법들은 몇 가지 기준에 의해 분류될 수 있다; 1) 부분 특유 방법 vs. 표현적 방법, 2) 3D 자세 추정 방법 vs. 3D 자세 및 형태 추정 방법, 그리고 3) 단일 사람을 위한 방법 vs. 여러 사람을 위한 방법. 부분 특유에서 표현적으로, 3D 자세 추정에서 3D 자세 및 형태 추정으로, 단일 사람에서 여러 사람으로 갈수록 추정이 더

어려워지지만, 더 풍부한 정보를 출력할 수 있게 된다.

본 학위논문은 단일 이미지로부터 여러 사람의 표현적인 3D 자세 및 형태 추정을 향하는 세 가지의 접근법을 소개한다. 따라서 최종적으로 제안된 방법은 가장 풍부한 정보를 제공할 수 있다. 첫 번째 접근법은 여러 사람을 위한 3D 자세 추정이고, 두 번째는 여러 사람을 위한 3D 자세 및 형태 추정이고, 그리고 마지막은 여러 사람을 위한 표현적인 3D 자세 및 형태 추정을 위한 방법이다. 각 접근법은 기존 방법들이 가진 중요한 한계점들을 해결하여 제안된 방법들이 실생활에서 쓰일 수 있도록 한다.

첫 번째 접근법은 여러 사람을 위한 3D 자세 추정 프레임워크이다. 단일 사람의 경우와는 다르게 여러 사람의 경우 사람마다 카메라 상대적인 3D 위치가 필요하다. 카메라 상대적인 3D 위치를 단일 이미지로부터 추정하는 것은 매우 높은 깊이 모호성을 동반한다. 제안하는 프레임워크는 심층 이미지 피쳐와 카메라 핀홀 모델을 사용하여 카메라 상대적인 3D 위치를 복원한다. 이 프레임워크는 어떤 단일 사람을 위한 3D 자세 및 형태 추정 방법과 합쳐질 수 있기 때문에, 다음에 소개될 두 접근법은 오직 단일 사람을 위한 3D 자세 및 형태 추정에 초점을 맞춘다. 다음에 소개될 두 접근법에서 제안된 단일 사람을 위한 방법들은 첫 번째 접근법에서 소개되는 여러 사람을 위한 프레임워크를 사용하여 쉽게 여러 사람의 경우로 확장할 수 있다. 두 번째 접근법은 여러 사람을 위한 3D 자세 및 형태 추정 방법이다. 이 방법은 첫 번째 접근법을 확장하여 정확도를 유지하면서 추가로 3D 형태를 추정하게 한다. 높은 정확도를 위해 릭셀 기반의 1D 히트맵을 제안하고, 이로 인해 기존에 발표된 방법들보다 큰 폭으로 높은 성능을 얻는다. 마지막 접근법은 여러 사람을 위한 표현적인 3D 자세 및 형태 추정 방법이다. 이것은 몸, 손, 그리고 얼굴마다 3D 자세 및 형태를 하나로 통합하여 표현적인 3D 자세 및 형태를 얻는다. 게다가, 이것은 3D 위치 포즈 기반의 3D 회전 포즈 추정기법을 제안함으로써 기존에 발표된 방법들보다 훨씬 높은 성능을 얻는다.

제안된 접근법들은 기존에 발표되었던 방법들이 갖는 한계점들을 성공적으로 극복한다. 광범위한 실험적 결과가 정성적, 정량적으로 제안하는 방법들의 효용성을 보여

준다.

주요어: 3D 인간 자세, 3D 인간 형태, 표현적 전신, 여러 사람, 단일 이미지

학번: 2015-22785

감사의 글

먼저 5년 반 동안 부족한 저를 성심성의껏 지도해주신 이경무 교수님께 진심으로 감사드립니다. 학위 기간 동안 연구적인 측면뿐 아니라 학자로서 갖추어야 할 태도에 대해서 이경무 교수님께 정말 많은 것을 배웠습니다. 국내뿐 아니라 세계 컴퓨터 비전 연구자 중에서 최고의 위치에 계심에도 불구하고 항상 열정을 가지고 겸손한 모습으로 진지하게 연구에 임하시는 모습이 열정과 패기가 가득했지만 침착함, 냉철함, 그리고 겸손함이 부족했던 저에게 큰 자극과 가르침이 되었습니다. 앞으로도 이경무 교수님을 본받아서 연구자로서의 삶을 성실히 살아가도록 하겠습니다.

포항공과대학교 컴퓨터공학과에서 학부생으로 공부하던 시절, 저의 학부 졸업 후 진로에 대해서 성심성의껏 상담해주신 한보형 교수님과 최승문 교수님께 감사드립니다. 한보형 교수님의 컴퓨터 비전 수업을 듣고 컴퓨터 비전을 연구해야겠다고 마음을 먹게 되었고, 그 결과 컴퓨터 비전을 연구하는 연구자가 되었습니다. 뿐만 아니라, 학부 졸업 후 컴퓨터 비전 연구실에 진학하고 싶었을 때, 서울대학교 이경무 교수님 연구실을 소개해주셔서 제가 이경무 교수님의 연구실에 입학할 수 있었습니다. 최승문 교수님의 인간-컴퓨터 상호작용 수업을 듣고 이 분야, 혹은 인간과 관련된 주제로 연구를 하고 싶다고 생각했습니다. 그 결과 3D 인간 자세 및 형태 추정이라는 주제로 박사학위 논문을 작성하게 되었습니다. 비록 인간-컴퓨터 상호작용 연구실로 진학하진 않았지만, 학부를 졸업할 당시 연구자가 가져야 할 태도 등에 대해서 성심성의껏 상담해주셔서 감사드립니다.

포항에서 25년을 살다가 낯선 서울에 올라왔을 때, 연구실 사람들은 저의 좋은 친구

가 되어주었습니다. 특히, 연구실 선배들은 저에게 많은 것을 가르쳐준 고마운 분들입니다. 제가 연구실 인턴일 때, 저의 많은 질문에 모두 잘 답해주셨던 희수형과 명섭이에게 감사합니다. 그리고 신입생 교육 때 많은 신입생에게 항상 친절하게 교육을 진행해 주었던 해솔이 형에게도 감사합니다. 제가 저년차일 때 가장 많이 배우고 연구자로서의 예리함에 놀랐던 유민 누나에게 감사합니다. 비록 연구주제가 달랐지만 몇 번의 co-work의 기회가 유민누나와 있었는데, 그 시간이 저에게 정말 큰 배움이 되었습니다. 특히 유민 누나의 예리함에 제가 허를 찔릴 때가 많았고, 그것이 저에게 큰 자극과 가르침이 되었습니다. 이외에도 푸근한 매력을 가진 정민이 형, 지금은 한양대학교 교수로 재직하고 있고 본받을 게 많은 형인 태현이 형, 마성의 매력을 가진 광모 형, ICCV 2017에서 많이 친해졌던 승연이 형, 걸어 다니는 나무위키 장훈이 형, 코로나와 이사, 그리고 결혼 때문에 보기가 힘들어진 의영이 형, 항상 성실히 연구실에 나와서 열심히 연구하는 승준이, 상남자 동우, 그리고 스타트업을 하는 지홍이 형에게 모두 감사합니다.

저의 하나뿐인 연구실 동기 성용이에게도 감사합니다. 성용이는 제가 낯을 가리고 다가가기 어려운 타입인데도 항상 먼저 말을 걸어주고 친하게 지내주었습니다. 그뿐만 아니라 연구실 대부분의 사람과 친하게 지내어 연구실의 분위기를 화기애애하게 만들어줍니다. 아마 연구실의 모든 사람이 성용이를 좋아하리라 생각합니다. 그것 때문인지 성용이가 지금 방장을 맡고 있는데, 앞으로도 맡은 방장의 역할을 잘 해내 주길 바랍니다.

짧지 않은 5년 반 동안 연구실에 있으면서 후배도 많아졌습니다. EDSR을 쓰고 이젠 스타트업을 하는 비, 학회에서 가끔 만나는 준형이, 특유의 친화력으로 많이 친해졌던 병건이 형, 교수님의 애제자 상현이, 후배들과 친하게 지내는 희원이형, 외국에서 와서 외로울 텐데 잘 지내고 있는 Mohsen과 Reyhaneh, 전혀 동생 같지 않은 재영이, 의영이 형과 함께 연구실에서 얼굴 보기 가장 힘든 석일이, 내 옆자리에서 고생하고 있는 현진이, 이번에 석사 졸업하게 된 건운이, 입대해서 지금 군 생활 열심히 하고 있을 재희, 듬직한 수영이, 연구실 운동 왕 영욱이, 추운 겨울에 입대해서 고생하고 있을 재린이,

super-resolution의 희망 재하, 얼굴 보기 힘든 강건이, 같은 분야 연구하는 파이터 홍석이, 친화력 좋고 엉뚱한 매력이 있는 채은이, 어몽어스와 인디언 포커 고수 정훈이, denosing의 희망 우석이, 자리가 외진 곳에 있어서 보기 힘든 도희에게 직/간접적으로 받은 도움에 대해 감사합니다. 연구실에서 혼자 3D 인간 자세 및 형태 추정 분야를 연구하다가 홍석이가 입학한 후 같이하고 있는데, 이 기회를 통해 후배와 일하는 법에 대해서 많이 배웠습니다. 처음에 후배와 같이 일하는 법을 몰랐을 때 약간 벅찰 수 있는 기준을 많이 요구했는데, 그것에 맞추기 위해 열심히 연구해준 홍석이에게 특히 감사합니다. 앞으로도 같이 좋은 연구들 많이 하면 좋겠습니다.

박사과정 초년 차 때 같이 co-work을 많이 했었던 장주용 교수님께 감사드립니다. 아직 다듬어지지 않고 서툰 게 많았던 시절, 장주용 교수님의 지도를 받아 더 나은 방향으로 나아갈 수 있었습니다. 학회를 통해 처음 뵙고 인사를 드리다가 co-work을 하게 되었고, 그 후 저에게 Google Fellowship 추천서를 써주시는 등 큰 도움을 주고 계신 조민수 교수님께 감사드립니다. 조민수 교수님과의 co-work은 제가 한 단계 더 성장할 수 있는 큰 가르침이 되었습니다. 2019년 Facebook Reality Lab, Pittsburgh에서 research intern으로 재직할 때, 저의 manager였던 Takaaki Shiratori, co-work을 했던 Shoo-i Yu와 He Wen에게 감사합니다. 피츠버그에서의 시간은 저에게 정말 큰 의미가 있었습니다. 연구적으로 큰 성장을 하게 되었고 연구와 연구 외적인 면을 모두 포함하여 더 넓은 시야를 가지게 되었습니다.

부모님과 동생을 포함한 가족들에게 감사합니다. 언제나 내 편을 들어주고 응원해 주는 가족이 있어서 오직 연구에만 집중하여 성과를 낼 수 있었습니다. 저도 언젠가 부모가 된다면 저의 부모님처럼 자식에게 든든한 버팀목이 되어줄 수 있는 부모가 되겠습니다. 그리고 사랑하는 여자친구이자 곧 아내가 될 윤아에게 감사합니다. 연구하느라 많은 시간을 같이 보내지 못해도 이해해주고 칙칙할 수 있는 저의 삶에 밝음을 불어넣어 주는 고마운 존재입니다. 앞으로 결혼을 하고 평생을 같이 재미있게 살아가겠습니다.

이외에도 포항공과대학교 10분반, 컴퓨터공학과, 브레멘, RA 등에서 알게 된 모든

사람에게 감사합니다. 또한 요즘에도 가끔 연락하는 고등학교 친구들인 종권이, 형탁이, 그리고 저의 재수 친구 요한이에게 감사합니다. 마지막으로 할아버지, 할머니, 외할머니, 돌아가신 외할아버지와 모든 친척분께 감사드립니다.