UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

3D Human Pose Estimation in Different Environment Settings Using Deep Learning Methods

by

Congzhentao Huang

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2023

Certificate of Original Authorship

I, Congzhentao Huang declare that this thesis is submitted in fulfillment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 12/05/2023

ABSTRACT

3D Human Pose Estimation in Different Environment Settings Using Deep Learning Methods

by

Congzhentao Huang

Three-dimensional human pose estimation methods have received widespread attention in the field of computer vision, from which many related applications have been derived. Such technology can estimate 3D human skeletons in the real world from camera images. Although early methods used images from a monocular camera to train a neural network, such approaches suffered from ambiguous depth and self-occlusions. Researchers, therefore, began to explore a multi-view approach to alleviate such problems. As another challenge, in most studies, massive numbers of labeled training data have been required for network training. Annotating 3D poses as the ground-truth using traditional marker-based motion capture systems is an expensive process. Hence self-supervised methods have attracted significant attention because network training can be conducted using only weak or even no supervision instead of applying paired 2D-3D human pose annotations.

We explore several methods using different parameters, such as a monocular camera, multiple individuals captured from multiple cameras, and a collaborative estimation using both cameras and radar. Three settings are related closely. The first setting is a multi-view multi-person detector. While it performs well, it needs lots of ground-truth data to train the network. Then we begin considering the second setting which does not need the labeled data for training. Then we begin to think about the drawback of the camera-based network, the solution is to add another kind of data, which is radar signals, to help train a more robust network. Three methods are described as follows:

1. We propose a novel end-to-end training scheme for multi-view multi-person 3D pose estimation. Our model back-propagates the gradients from the last 3D estimation step to the first 2D detection step, thereby significantly improving the efficiency, robustness, and accuracy of the 3D pose estimation. We also designed a multi-view 2D human pose dynamic matching algorithm, which can dynamically match the corresponding 2D poses detected in multiple views for each person involved.

2. We propose a two-branch self-supervised approach in a multi-view training setting to train a 2D-3D neural network without the use of 3D ground-truth labels. The entire model only relies on geometric information in the building of supervision signals.

3. We propose a novel unsupervised model that infers 3D human skeletons from radar signals. This method solves such problems as poor illumination, adverse weather conditions, or occluded body parts, which affect the camera, by training the network using both camera images and radar signals.

Acknowledgements

First and most sincerely, I would like to thank my supervisor, Professor Andrew Zhang. Andrew is very hardworking, and kind, and strives for excellence, which has led me to learn from him continuously. In the past few years, he taught me knowledge and skills for academic research. There is a famous saying in ancient China, "In ancient times those who wanted to learn would seek out a teacher, one who could propagate the doctrine, impart professional knowledge, and resolve doubts." I think Andrew fits that quote perfectly. Thank you, professor. I also would like to thank my co-supervisor Richard Xu, Sam Ferguson, and Jay Guo, who led me into the research work for the first time.

I also want to thank my fellow labmates: Ziyue Zhang, Yang Li, Chen Deng, Wei Huang, Shuai Jiang, Haodong Chang, Xuan Liang, Ximeng Zhao, Caoyuan Li, Jason Traish, etc. We discussed the experiment details and innovations of each paper. We work together before conference deadlines sleeplessly. We also enjoyed the hotpot and movies on the weekend. We shared joys and pains from life and research. It is my pleasure to have had such good friends in the past few years and the future.

I would also like to thank the officers from GRS and SEDE of UTS. During the past years, they have helped me a lot with the admin and research progress. Thanks so much for your patience and kindness.

Sincerely, I must thank my parents, Donglin Huang and Zhihong Zhou. When I was young, you taught me to be a hardworking and sincere person. I could not have gotten through those struggling times without you by my side. Your continuous encouragement and support are my magic bullets over those difficult problems.

Congzhentao Huang

List of Publications

Journal Papers

J-1. Zhang Z., Jiang S., **Huang C.**, Li Y., Xu R. Y. D. (2021). RGB-IR cross-modality person ReID based on teacher-student GAN model, Pattern Recognition Letters, 150, 155-161.

J-2. Zhang Z., Jiang S., **Huang C.**, Xu R. Y. D., Unsupervised clothing change adaptive person ReID, IEEE Signal Processing Letters

Conference Papers

C-1. **Huang C.**, Jiang S., Li Y., Zhang Z., Traish J., Deng C., Ferguson S., Xu R. Y. D., End-to-end Dynamic Matching Network for Multi-view Multi-person 3d Pose Estimation, accepted by ECCV, 2020.

C-2. Zhang Z., Jiang S., **Huang C**., Xu R. Y. D., Resolution-Invariant Person Reid Based On Feature Transformation And Self-Weighted Attention, 2021 IEEE International Conference on Image Processing (ICIP), 2021.

C-3. Zhang Z., Xu, R. Y. D., Jiang S., Li Y., **Huang C**., Deng C., Illumination Adaptive Person ReID based on Teacher-Student Model and Adversarial Training, accepted by ICIP, 2020.

C-4. Li Y., Li K., Jiang S., Zhang Z., **Huang C.**, Xu R. Y. D., Geometry-driven self-supervised method for 3D human pose estimation, In Proceedings of the AAAI Conference on Artificial Intelligence 2020.

Submitted and to be Submitted Papers

1. **Huang C.**, Zhang Z., Xu R. Y. D., Zhang A., Self-supervised Network for 3D Human Pose Estimation, submitted to WACV, 2022.

2. **Huang C.**, Pearce A., Xu R. Y. D., Zhang A., 3D Human Pose Estimation with mmWave Radar, to be submitted.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Deep learning is an essential part of modern computer vision. From basic tasks such as object recognition[100, 45], object detection[43, 91] and object tracking[122, 121], to advanced semantic tasks including traffic scene understanding [38, 89], the community has witnessed significant performance gains in such algorithms. This is due to the advanced GPU hardware providing powerful computing power for rapid prototyping and various applications. Although the deep learning method has proved to have a generalization ability far exceeding that of many artificial designs, it is only comparable to the universality of the training data themselves. In terms of cost and time, data annotation is increasingly becoming the most expensive item in the development of such algorithms. There are currently many ways to address the true lack of dataset availability. Most notable is the development of computer graphics simulation technology, which excels in producing real images, point clouds, CAD objects, and more. Open-source software such as Blender and NVIDIA Deep Learning Dataset Synthesizer have been widely used for generating images of objects. In addition to synthesizing data-generating software, pioneering research in the form of a generative adversarial network (GAN) [40] can also synthesize "real" pseudo-data by learning the intrinsic representation of existing real-world datasets. With the active development of data collocation, human pose estimation has been successfully fueled by recent deep-learning techniques.

In the field of computer vision, human pose estimation is the task of locating

important joints in the human body. The skeleton can be properly described by estimating the human key-points, which can assist with other tasks in similar areas, such as activity recognition and object tracking. The main purpose of this study is to explore the estimation of 3D human poses using a convolutional neural network (CNN). This chapter provides a reference for this research and describes the organization of the present paper.

1.2 Aims and Motivations

Pose estimation is a major research direction in both computer vision and machine learning. It is widely used in film production, human-computer interaction, and other fields. In these fields, the detections can be very helpful for management or professional use. In addition, in other similar areas, such as image recognition, the results of the pose estimation can be helpful. For both commercial and academic purposes, many videos are produced and must be processed and analyzed, in real-time. Accurate, fast, and robust RGB-video-based 3D pose estimation technology will greatly facilitate the realization of the above goals. In recent years, with the rapid development of CNN technologies, pose estimation approaches have been developed. The use of a CNN is presently the main research direction in the field of image processing. At the same time, the method has also reached significant achievements in pose estimation. It can be said that most of the current pose estimation algorithms are based on a CNN.

There are many problems in the field of pose estimation, like lack of ground truth data and occlusion problems. The focus of this thesis is the estimation of a 3D pose. In 3D pose estimation, many cases rely on an estimation of a 2D pose. A video can be viewed as a series of images in RGB, and therefore a pose estimation in the collected images should first be completed. Finally, by synthesizing and smoothing the pose estimation results of each frame, the pose estimation value in the RGB

image is obtained. In a 2D pose estimation, the estimated key points are expressed based on the coordinates of important nodes in the image. In a 3D pose estimation, we consider both visible and self-occluded joints.

Although CNN-based pose estimation technology has made significant progress, there are many shortcomings to existing RGB pose estimation algorithms, such as a lack of training data, an inability to generalize, and the accuracy of the 3D pose estimation.

To discuss ways to solve the above problems in detail, this thesis has been structured into the following chapters. We focus on three research areas:

1. **Objective 1**

   Current studies on multi-view multi-person 3D pose estimation have difficulties performing well in terms of both accuracy and effectiveness. In Chapter 3, We propose a novel end-to-end training scheme for multi-view multi-person 3D pose estimation. Differing from the separate training of independent modules, our model back-propagates the gradients from the last 3D estimation step to the first 2D detection step, so as to thereby significantly improve the efficiency, robustness, and accuracy of a 3D pose estimation. A multi-view 2D human poses dynamic matching algorithm is also proposed. This algorithm can dynamically match the corresponding 2D poses detected in multiple views for each person involved. The approach does not require knowing the exact number of people on the scene and can handle cases in which false detections and severe occlusions occur. Experiments on the Shelf and Campus datasets demonstrate that our proposed model outperforms other state-of-the-art approaches in terms of both efficiency and accuracy.

2. **Objective 2**

   Most previous studies in this field have relied heavily on a large open dataset

containing both 2D and 3D ground-truth annotations. In chapter 3, we use ground-truth data to train a multi-view, multi-person network. Next work we consider using a self-supervised network. In chapter 4, we propose a two-branch self-supervised approach in a multi-view training setting for training a 2D-3D neural network without 3D ground-truth labels. The entire model only relies on the geometrical information for the building of supervision signals. The model is trained using a cycle-view training scheme, which is effective in exploiting multi-view consistency and constraining the 3D estimations during the training stage. The method overcomes the depth ambiguity problem and can handle incomplete or false 2D detections by utilizing the information from other views. Moreover, to solve the occlusion problem, we make use of the 2D joint confidence from different cameras. Evaluations of the Human3.6M and MPI-INF-3DHP datasets demonstrate that our proposed model achieves state-of-the-art results compared with recent self-supervised methods.

3. **Objective 3**

Two problems remain in a self-supervised approach to 3D human pose estimation. First, camera images are easily affected by the lighting conditions, and occlusions are difficult to overcome in certain experiments. Second, although self-supervised training does not require the ground-truth data, a camera system is still needed to capture large numbers of 2D images. To solve these problems, we use both a camera and a radar device to collect a small dataset used for network training. Last two chapters we use only camera images to train the network. In chapter 5, we propose a two-stream self-supervised approach to extracting 3D skeletons and their key-points from radar signals. The model is trained using a mix of camera and radar data, which enhances the robustness of the network. The network is designed in a self-supervised manner, which means the model does not require a ground-truth label during the

training. The camera and radar streams are both used during the training stage, whereas only the radar signals are used in the evaluation. We collected a small dataset for the training and generated 3D labels for the evaluation using multi-view algorithms. Our evaluations of the collected data demonstrate both the effectiveness and robustness of our approach.

1.3 Thesis Structure

This thesis mainly focuses on human pose estimation based on deep learning methods, and is organized as follows.

- *Chapter 2*: In this chapter, we present the theoretical background of the present thesis. In particular, we describe previous studies on image processing techniques and human pose estimation related to our approach.

- *Chapter 3*: In this chapter, a novel end-to-end training scheme for multi-view multi-person 3D pose estimation is proposed. A multi-view 2D human pose dynamic matching algorithm designed for identifying people in different camera views is also described.

- *Chapter 4*: In this chapter, a two-branch self-supervised approach used in a multi-view training setting for training a 2D-3D neural network without the use of 3D ground-truth labels is presented. The entire model only relies on the geometry information for building supervision signals. This chapter explores the use of 3D pose estimation without labels.

- *Chapter 5*: In this chapter, a two-stream self-supervised approach for extracting 3D skeletons and their key-points from radar signals is presented. The model is trained using a mix of camera and radar data, enhancing the robustness of the network. This chapter is a further exploration of the no-label

training of a 3D human pose, which applies the advantages of radar to enhance the generalization ability of the system.

- *Chapter 6*: In this chapter, a summary of the contents of this thesis and our directions for future research are provided.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, an overview of previous research conducted on an attitude assessment is provided. The methods and concepts introduced herein are the main contents of the later chapters in this paper. Pose estimation is one of the branches in the field of image processing. On this basis, the original image is extracted using an image processing method for further use. Section 2.2 describes some of the fundamental techniques of image processing. Section 2.3 presents previous methods used for pose estimation.

2.2 Image Processing

Image processing is the basic but important step in image-based studies. In pose estimation, early image cutting, detection is an important step in all studies. The following sections will introduce some important image-processing technologies which are widely used in pose estimations.

2.2.1 Object Detection

Object recognition is extremely important in the field of image processing. On this basis, a task can be divided into two categories. One is an area proposal task, which aims to identify areas that may be targeted. The second is classification according to whether the proposed area contains specific objects.

As the greatest difficulty in object detection, objects are extremely different in

terms of scale, illumination, and viewpoint, and the materials and poses are quite different [34]. A method having the ability to detect one type of object may not detect another type of object. In addition, the detection of large targets is not necessarily suitable for the effective detection of small targets.

In recent years, research on object detection has mainly focused on deep learning-based algorithms. On this basis, a new deep learning-based object recognition algorithm is proposed. Currently, the most commonly used object detection datasets are the PASCAL VOC 2012 [29], ImageNet [22], and COCO [73] datasets. The ImageNet dataset includes over 150,000 labeled categories, whereas the COCO dataset contains 200,000 images with object segmentation information.

At present, many object recognition technologies based on deep learning have been applied, the basic architectures of which can be divided into region-proposal and regression/classification-based methods.

Region-proposal based methods

On this basis, an object recognition model based on a region proposal is proposed, which can be divided into two major blocks. Before proposing a region-based convolutional net method (RCNN) [39], object detection techniques employ a sliding window as a region proposal, which scans the entire image once. Conducting operations under a large number of sliding windows will generate a large number of calculations, resulting in a slower operational speed.

The RCNN method employs a "selection search" technique for extracting region recommendations from the categories listed below, thereby significantly reducing the computational area to approximately 2000 regions. The selective search method divides the image into several small regions, then iterates according to the color space, combines them with other similarity measures, and outputs the objects within the 2000 regions. An RCNN also employs a CNN for feature extraction and classification

of these regions. There are three major problems inherent to an RCNN architecture. First, the selective 2000 region area can occupy a large number of hard disks. Second, to provide data to the CNN, regions of different sizes need to be cropped to the same size, which will result in a loss of information. In addition, it will take a while for these 2000 region proposals to be processed individually by CNN.

SPP-Net [44] improves upon the RCNN method. The algorithm computes the features only once for the entire image. SPP-Net employs a spatial pyramiding (SPP) model to normalize the features proposed by the regions, which reduces the computational time. However, as with an RCNN, the selection search model used in SPP-Net is less effective in terms of time and requires a large amount of disk space.

Fast RCNN [39] is based on the SPP-Net and RCNN methods. This method introduces the software Max function in the CNN classifier and replaces the SPP with the ROI module. A new CNN method based on VGG16 has also been proposed. This algorithm also adopts a multi-task loss approach to achieve the end-to-end learning effect. Compared with SPP-Net, an RCNN is faster and more accurate. However, the bottlenecks of Fast RCNN remain the time and disk consumption of the search region proposal module.

Faster RCNN [32] solves the above problems by applying a region proposal neural network. It first conducts feature mapping on all input images and then uses the region proposal neural network for regression. Next, the returned region proposals are finalized and localized using the ROI pooling layer of interest. Faster RCNN is an extremely fast algorithm. It can also be applied to the real-time detection of objects. However, it is difficult for Faster RCNN to detect smaller objects owing to the fixed-scale anchor box mechanism.

R-FCN [19] was developed based on Faster RCNN [32]. This method uses the structure of a fully convolutional neural network. Finally, the extracted features

are input into an RPN, such as the regression ROI used in Faster RCNN. The next feature in the ROI is then fed into the score map network for classification. Compared with Faster RCNN, R-FCN achieves higher accuracy and speed. However, the detection of small targets remains problematic.

A feature pyramid network [72] was developed for overcoming the small object detection problem with Fast RCNN. ResNet is used as a basic building block in an FCN. First, the feature maps with different scales are cone-shaped. Finally, the features of different scales are combined to make the final forecast. Because the information of various scales is stored in the final forecast, the FCN is better able to detect smaller targets.

Mask RCNN [43] was developed based on Faster RCNN. It replaces the pooling layer with an aligned layer to provide region proposals with equal variance and equal variance. Feature extraction also adopts ResNet 101 and an FPN. In addition to object classification and object localization, it also adds an FCN layer, which adds a mask generation behind the RoIAlign layer. The FCN layer is parallel to the other two types and location hierarchies.

Regression Methods

A regression algorithm adopts a one-step point-to-point approach to achieve the overall localization and classification of objects, which is extremely different from the design of the two models based on the region proposal. This method is simpler and faster. The main methods include YOLO [91] and SSD [105].

The YOLO method divides the input image into 77 sub-regions. In each subregion, a fully connected layer is used to regress the rectangular boxes, which contain the center of mass and the confidence score. This confidence score is a product that includes the target probability and the intersection over union (IoU) between the bbox and the ground truth. The network is designed based on GoogleNet [105] and

is deeper, with the inclusion of an inception module. Regressed bounding boxes are merged using a non-maximum compression method. Owing to the simplicity of the network, the speed of the Yolo approach is greatly improved. However, the accuracy is relatively lower compared with that of the other models.

SSD is another method that makes use of a convolutional neural network. The network takes inputs of the images and then feeds them into the layers with different filter sizes. Feature patches are then predicted using an additional CNN layer. Each block has a center coordinate, width, height, and probability of containing all object classes. Finally, the non-maximum suppression methods are combined for the final prediction. The SSD algorithm has a high calculation accuracy and does not affect the operational speed.

The performance of different object detection methods was evaluated using the mean accuracy (mAP), which is a common method for measuring the accuracy of a target detection algorithm. In object detection, an edge box and a class marker are applied. If the class label is the same as the ground-truth, and the IoU between the prediction boundary and the ground-truth boundary block exceeds a threshold (typically 0.5), it will be marked as an actual prediction. In object detection, a forecast means that the actual forecast accounts for a part of the total forecast results and is used to measure the accuracy of the forecast. The backtracking value is the actual prediction score for all ground-truth objects and is used to measure the number of all detected frontal objects. Unfortunately, this accuracy is inversely proportional to the recovery value. To balance the relationship between the two and evaluate the performance, mAP is introduced.

2.2.2 Object Tracking

Object tracking is the next stage after object understanding. Target tracking refers to locating the target in a continuous image sequence to obtain the entire

motion trajectory. First, the position of the target must be determined, and the positioning of the next frame can then be carried out.

Object tracking is similar to an object detection task over time. However, it differs in that it requires locating the same object of different frames over time. In addition, because target tracking requires a high computing speed, the algorithm always finds a balance between speed and accuracy.

Object tracking technology has problems such as a deformation, change in illumination, motion blur, background clutter, and occlusion problems. These problems make it difficult to track targets accurately and quickly.

Object tracking has long been a studied technique. When a machine learning algorithm is fully developed, the new tracking algorithm will greatly improve the accuracy and speed. Such methods are divided into two broad categories: generative and discriminative approaches, which are introduced next.

Generative Methods

Generative methods first extract the motion area from two adjacent images and then identify and finally determine the target. Therefore, with this algorithm, motion detection is the first step in tracking the target. Motion detection generally extracts the changing range of an image from a static background. To solve this problem, the optical flow method is generally applied.

Optical flow refers to a dynamic caused by the relative movement between the target and background in a continuous image. A calculation of the optical flow can be conducted in a number of ways. Among such approaches, the Lucas-Kanade method [74] is the most widely used method.

The Lucas-Kanade method is based on two assumptions. As the first assumption, the color of an object barely changes between two adjacent frames. As the

second, there is little relative motion of objects between two adjacent frames. In practical applications, the Lucas-Kanade algorithm cannot effectively solve large-scale movements and changes in lighting owing to its basic assumptions. By adopting an iterative pyramid and sparse optical flow method, the operational speed of the algorithm is improved, and a large number of relative movement problems are solved.

Other generative methods include mean-shift [18], Camshift [7], and Kalman filter [61] approaches. These algorithms construct the pattern of the target area according to its color and the characteristics of the current image and then find the closest image in the next frame. Among them, the ASMS method [112] achieves the best results. This method adds a scale estimate to the standard mean deviation frame, enabling it to reach a frame rate of 125 fps.

Discriminative Methods

Recent discriminative-based algorithms perform better than generative algorithms. Compared with traditional recognition algorithms, a discriminative-based algorithm uses traditional templates in the processing of the image features. The algorithm takes the target and background areas as a positive sample and a negative value for classification. This classifier is then used to detect the old position in the next frame. With a discriminative-based method, both the block of the target image and the information of the background image are used for tracking. Therefore, such a method is generally better than a production-based method. The idea of this identification is to trace through a detection.

Two early discriminative methods are TLD [60] and Struck [42]. TLD is focused on long-term tracking. TLD includes three parts: tracking, learning, and detection. The algorithm uses the traditional optical flow method with target detection, which effectively overcomes the deformation and obstacles during the target tracking pro-

cess. At the same time, the parameters of the tracking and detection modules are updated through an online iterative learning algorithm, which makes the tracking performance of the system more stable, robust, and reliable. The accuracy of TLD is 30 fps with a mAP of 42.5%. Struck uses Haar-like features and structural support vector machines to classify the images. The algorithm also applies an overlay-based sampling algorithm. It can achieve a frame rate of 20 fps with a mAP of 46%.

Correlation filter (CF) methods have improved effectiveness compared with previous methods. The author of [47] conducted the earliest research. A CF is implemented by finding a filter for determining the similarity between the target and the next image frame, and the Fourier transform is used to speed up the operation. Other methods are then proposed [6, 48, 21]. By training the filter with different features and scales, these methods can better reflect how similar two images are. More features are introduced, or larger scales can be more accurately tracked. However, this comes at the cost of a relatively slower speed.

In general, CF methods are faster than other approaches. However, the speed limit of the CF algorithm is mainly based on the update of the model and the sampling of the training set. Although each frame of the model update increases the accuracy, the speed decreases. Sampling only the end and end of the data also has an impact on the performance of the application.

In recent years, with the development of deep learning methods, some new methods have been proposed. The authors of [46] use a deep learning model to train the filter. This is the first tracking architecture based on end-to-end deep learning methods. Its speed reaches 100 fps, far exceeding that of other deep learning methods.

ECO [20] is another deep learning-based algorithm that achieves a high accuracy. To improve the operational speed, a decomposed convolution operator is used. On this basis, a new model is established to ensure the diversity of the samples. In

addition, to solve the drift of the model, ECO updates the model every 6 frames. As a drawback of ECO, it can only achieve a speed of 8 fps.

2.2.3 Activity Recognition

Action recognition is the process of recognizing various activities through video footage, in which actors perform various actions within a segment or video clip. This approach seems similar to a simple application of object detection and tracking to serial frames and generating a prediction in those images based on the results of the object tracking. However, although machine learning methods have achieved significant results in terms of object detection and tracking, an unsolved problem remains for the following reasons:

1. Diversity of the actions:

   The task of active identification is to capture the spatial and spatial characteristics of multiple frames. Owing to different environment settings, such as lighting conditions, obstacles, and resolution, as well as different people performing differently at different angles, backgrounds, or on different stages, for the same activity, the features extracted from the video will also vary. Plus, in a video clip, it is difficult to tell where the action begins and ends. The problems mentioned above will have an impact on effective video recognition.

2. Huge computational costs:

   Compared with traditional two-dimensional image processing methods, multi-frame video feature extraction requires the use of three-dimensional convolutional networks to solve the problem of many learnable parameters and long-time consumption in motion recognition.

3. Lack of a benchmark:

The motion recognition method is based on machine learning technology, which requires numerous images or video streams with different motions to train the network and achieve accurate predictions. Although some datasets have been developed, a problem remains for researchers conducting network training in complex settings.

Many methods have been proposed to solve these difficulties. Typically, the early methods of effective activity recognition [25, 69, 117, 117] were divided into three major steps: 1) extracting local high-dimensional features for a local region, 2) combining the extracted features into video-level features, and 3) classifying the results into the final predictions.

Among traditional algorithms, an improved compressive tracking (iDT) [117] algorithm based on density tracking is used, which achieves the highest results. The algorithm uses an optical flow field to track each frame. Then, using a grayscale image or dense optical flow, the HOF, HOG, MBH, and orbit of the three features are obtained. These features are then encoded using the Fisher vectorizer method. Finally, a support vector machine classifier is used to encode the features. This algorithm achieves good stability and robustness. However, the computational cost of the optical flow characteristics is high.

After a CNN is used for feature extraction and classification prediction, it is divided into single- and two-stream networks. In a single-stream network, only the spatial information of the frame is used for activity classification. The authors of [63] used several methods to combine the features extracted for the prediction. They import video clips into different CNNs. Under a single-frame structure, a CNN is used to process the frames separately, and the features of each frame are fused to predict the subsequent data. In the later stage, two networks with the same weight are adopted to process the frames of two fixed intervals individually and

fuse the extracted data. In the first stage, early fusion architectures fuse multiple consecutive frames. The slow fusion architecture adopts a new structure fusing the frames at different stages. In single-stream networks, most networks capture the spatial information of the image, which ignores the temporal information.

For the two-stream method, an optical flow is introduced. On this basis, a new algorithm based on temporal features is proposed. In the spatial background network, there is only a single video frame, and the optical flow is sent into the temporal stream. The flow of light contains information transmitted from several serial frames. In the next step, the extracted features are combined together to form a support vector machine for prediction. The approach in [99] first introduced the two-stream method. Although its performance is better than that of single streaming, it also suffers from many shortcomings, such as an erroneous label assignment and a large computational cost for optical flow predictions.

There are other methods developed based on above two methods. To capture the full dynamics, long short-term memory networks (LSTMs) are a natural choice, and can be used for long-term dependencies. A long-term recurrent convolutional network (LRCN) [26] is proposed for visual recognition and a description of the images. This method utilizes the LSTM approach to encode the data obtained for further prediction. In this study, various input methods such as RGB input, weighted RGB, and photocurrent input were compared. This paper presents a trainable end-to-end architecture. The results show that the temporal information and the estimation of the optical flow missed a lot.

C3D [111] captures the spatial and temporal properties of images by applying a 3D convolutional grid across the video. In this study, an extensive search was conducted. C3D combines iDT tracking with linear support vector machines, outperforming previous studies. However, for a long time period, the acquisition of time

information is still a difficult problem. On this basis, a factorable spatiotemporal convolutional network (FSTCN) is proposed [102]. In this paper, a 3D network consists of a 2D convolutional network and a 1D convolutional network, and the extracted features are merged into consecutive frames, with significant results obtained.

The authors of [33] proposed a new TSN algorithm that uses a two-step fusion. The first step is to combine temporal properties with spatial properties. In the second stage, the image is processed at multiple levels, combining the temporal and spatial properties of the post-processed features of the image to make the final prediction. This indicates that using soft attention techniques can effectively improve the performance of C3D with the same number of parameters.

TSN method [119] also uses a two-stream structure. The contribution of a TSN is the proposition of a method based on sparse sampling and in-depth research conducted on batch normalization and dropout techniques.

The I3D method [11] developed the C3D method. This paper introduced the 3D convolutional block model in both flow architectures. At the same time, it also uses a 2D convolutional network pre-trained on ImageNet. T3D [23] extends I3D with a 3D dense block based convolutional neural network and a transformation layer structure. The research also introduced a new technique called "supervision transfer learning," which is used between 2D convolutional networks and T3D networks. Both of them take input from the same image or different images from video clips or video frames. Combining the outputs of the two networks, a differential prediction of 0/1 was conducted on the parameters of T3D. In this way, the knowledge from a pretrained 2D network can be transferred to a T3D network.

2.3 Pose Estimation

In studies on computer vision, a human body posture assessment is an important aspect used to determine the various important joints of the human body. By locating the important joints of the human body, the skeleton of the human body can be properly described, thereby providing more references for other computer vision tasks. However, there are several problems with an attitude assessment.

1. Various human postures: The human body is soft, flexible, and diverse. People can make different gestures. Every joint of the body is involved in six different movements. The same person poses differently. In addition, owing to such factors as height and weight, different people will achieve a different performance when they conduct the same action.

2. Difficulty in setting the environment: Under a low image resolution, strong illumination, and different viewpoints, pose estimation is difficult owing to such factors as the scale of the image, the change in viewing angle, and a complex background. A pose estimation method that addresses one environment setting cannot be applied to other environment settings. In addition, in many images, some parts of the body are hidden. Therefore, it has been estimated that this link will be more difficult to apply.

Traditional pose estimation uses human templates for image matching. Artificial templates are based on what is known. Pictorial structures [35] are a traditional method for estimating the pose, which consists of two types: a unary template representing different parts of the body, and a pair of springs that define the spatial relationship of the body based on known information. The use of mannequins cannot cover the variety and complexity of shapes. The introduction of machine learning in pose estimation has become a common design tool. This paper introduces a machine

learning-based pose estimation algorithm.

2.3.1 2D Single-person Pose Estimation

In a single image, a convolutional network is used to locate the important nodes of the human body. For example, in the Deep Pose method [110], a coarse-to-fine approach to directly output the coordinates is applied. A HeatmapsNet method, developed by Flowing Convnets [109], has thus been generally adopted. Compared to the CoordinatesNet method, Heatmap Net regresses a series of heatmaps representing joint detection probability maps. Each pixel reflects the probability of a particular junction on that pixel. Thus, the closer a pixel value is to 1, the more likely it is to be on that pixel, and vice versa. Therefore, in this heat map regression method, it is often necessary to input a set of Gaussian temperature curves of Ground truth nodes to calculate the loss. These pieces of training provide far more information than training on the coordinate grid, speeding up the training process and improving the accuracy of the estimates. In addition, a visualization of the heatmap network is easier to implement during training.

The two main changes to the heatmap approach are the use of convolutional pose machines (CPMs) [120] and a stacked hourglass network [82]. A multi-stage intermediate training approach is used in a CPM. In step 1, only RGB images are input for the heatmap regression. In the next step, the heatmap is returned from the previous step, along with the input raw RGB image. An intermediate loss is computed at each step to update the deep convolutional network. In addition, the perception range on the image is enhanced, allowing the network to understand the long-distance spatial limits between different joints, as well as allowing the mode to deal with obstacles. However, because of the multi-layer structure, this method is extremely slow.

The stack hourglass network uses a typical encoder-decoder architecture. This

method uses the residual method to store images in layers, which improves the multi-scale resolution of the images. Like a CPM, intermediary supervision is required at each link to prevent gradual changes. The stack hourglass algorithm achieves a higher accuracy and higher computing speed. Since it was first proposed, many personal pose estimation methods have used the stacked hourglass structure, such as structured feature learning [17], advanced Posenet [16], and CPF [125].

2.3.2 2D Multi-person Pose Estimation

A multi-person 2D pose estimation can handle more than two people simultaneously. This is not simply about finding the key hinges in the imagery, it is about which hinges belong to which individual. Such methods can be divided into two categories. The first category is "top-down methods" [15, 30, 55], and they use an object detection method to detect all people in the image and send them separately to a single 2D pose detector to obtain their corresponding 2D poses. In [55], the authors constructed a fully connected graph from a set of detected joint candidates of each person in an image and resolved the joint-to-person association and outlier detection by applying integer linear programming. In [30], the authors proposed a framework with three components for a pose estimation, which can extract a high-quality single-person region from an inaccurate bounding box. In [15], a two-part network structure was proposed where GlobalNet localizes the "simple" key-points and RefineNet deals with the "hard" key-points. For the second category, "bottom-up methods" are used to jointly label the part detection candidates and associate them with individuals using a matching algorithm [88, 9, 53]. The authors in [9] mapped the relationship between key-points into part affinity fields (PAFs), and then clustered the detected key-points into different 3D human poses. In [88], the authors interpreted the problem of distinguishing different people in an image as an integer linear programming problem and partitioned the part detection

candidates into identity clusters. On the basis of [88], the authors in [53] used a stronger part detector based on ResNet [45] and image-dependent pairwise scores, vastly improving the run time by applying an incremental optimization approach.

2.3.3 3D Pose Estimation

There are two main types of 3D pose estimation: one- and two-step methods. Two-stage approaches [12, 50, 62, 107, 107, 131, 130] first apply a 2D pose estimator to generate 2D skeletons, and then regress the 3D pose from the 2D poses. In [124] and [67], the simple lifting framework is extended through an adversarial learning strategy. The authors designed a multi-source recognition system to train a deep regression model to generate more anthropometric results. The sources of the identification tools are 2D heatmaps, depth maps, geometric descriptors, and RGB maps.

Coarse-to-fine volume prediction [85] is another two-stage approach. This method is represented by a 3D human body pose. At each node, the convolutional network is used to discretize the space around the object, and a regression of each voxel likelihood is then conducted. As one of the advantages of a volumetric representation, it transforms a nonlinear problem in 3D stereo regression into a more tractable discrete spatial prediction model. This method also applies a coarse-to-fine strategy to increase the robustness.

LCR-Net [96] also adopts a coarse-to-fine style to restore 2D and 3D multi-person poses. The system utilizes RPN technology and uses a set of 2D and 3D fixed positioning poses to obtain pose cues from the images. Next, the classifier is used to determine the connection between nodes. Finally, the 2D and 3D poses are refined using a regression algorithm.

A one-step method [113] returns the 3D pose to the image. For example, leveraging deep ResNet [45], VNect [78] was developed to process the real-time estimation

of a 3D human pose from single-view images. These methods all require extensive label training of a CNN. The approach in [77] is an extension of VNect, which implements the partial occlusion of the human body by VNect through the introduction of occlusion-robust location maps (ORLMs).

For a multi-view 3D pose estimation, traditional methods [2, 5, 8] have used a 2D pose estimation captured by calibrated cameras to predict 3D poses through a point triangulation or 3DPS. Recent studies have begun to adopt deep neural networks in this area and have delivered significant achievements. For example, in [57], a volumetric triangulation approach was proposed for projecting the feature maps produced by 2D pose estimators into 3D volumes, which were then used to predict the 3D poses. There are also self-supervised approaches that predict 3D poses separately in different camera views and minimize the distance between pairwise 3D poses after rotating toward the same view [64, 95, 14].

For a multi-view multi-person 3D pose estimation, 3DPS is the most widely used approach [3, 4, 59]. It predicts 3D key-points or 3D body parts by exploring an ample state space, and the candidates in the state space are generated through grid sampling. With the 2D priors given by the 2D detector, the 3D pose can be generated using maximum likelihood estimation. A recent study [27] proposed a model for combining person re-identification (re-id) [128, 129] and an epipolar geometry to match the pose, followed by the prediction of 3D poses using 3DPS. As a shortcoming of this approach, the speed of the person re-id model is relatively slow, which causes problems in terms of efficiency.

Chapter 3

End-to-end Dynamic Matching Network for Multi-view Multi-person 3D Pose Estimation

3.1 Introduction

3D human pose estimation is a fundamental problem in computer vision. It can be applied to various applications such as human-computer interactions, augmented reality and video surveillance. Due to the availability of increasingly sophisticated datasets, and more and more powerful deep learning models, researchers have made significant progress in this area using deep convolutional neural networks (CNNs). While 3D pose estimation research into a single human under monocular or multi-camera settings has made remarkable advances, fewer works have studied 3D pose estimation of multiple humans, which is a significantly more challenging problem to address. This is primarily due to the occurrences of frequent and sometimes severe occlusions when multiple people are involved. These difficulties have been further exacerbated by the lack of labeling for identifying corresponding people under a multi-view setting.

Despite these difficulties, there are two main reasons why multi-view multi-person 3D pose estimations will become mainstream research. First, models involving multiple people are more generic in many real-world applications compared to those for a single human, such as in supermarkets and factories. Secondly, using multi-cameras, the pose estimation can be made more robust than using a monocular camera due to the multiplied information available from different views, such as when dealing with occlusions.

The methodology for multi-view multi-person 3D pose estimation in many existing studies includes two steps. The first is to predict 2D poses in each view individually using off-the-shelf 2D models [9, 82, 15]. The second is to aggregate these 2D poses and generate their 3D counterpart. One typical idea is to use the so-called 3D Pictorial Structures model (3DPS), which directly generates 3D human poses by exploring an ample state space of all possible human key points or human body parts in 3D space [59, 4]. However, this method lacks efficiency due to the enormous state space needed for exploration.

In contrast to the above two-step models, a recent direction is to use a matching algorithm that identifies matched 2D skeletons from multiple views before the estimation for 3D poses [27]. If the matching algorithm is perfect, the subsequent 3D pose estimation for multiple people can be regarded as multiple 3D pose estimation for a single person. Thus the accuracy will be significantly improved. However, the matching algorithm may make mistakes or even fail. Once a reliable skeleton matching is established, we can then build an effective model in which its pipeline consists of three separate steps: (1) detect 2D skeletons in each camera view, (2) identify matched skeletons and (3) estimate the 3D pose.

An intuitive approach is, of course, to train each of these steps/modules independently. During testing, we can feed the 2D images and camera parameters through these trained modules one by one. However, all of the three operations are highly correlated in both directions of the pipeline. How individual poses are extracted in step 1 will undoubtedly influence the 3D pose estimation result in step 3. The reverse is also true: any adjustments that occur in the 3D estimation in step 3 will ultimately affect the way in which the detection should be carried out in step 1. Therefore, it is essential that the information can be back-propagated in reverse order through step 3 to step 1.

At the same time, when the parameters of the detection module in step 1 are not trained properly, especially during the early stage of the training, the matching algorithm in step 2 may fail to identify the matched skeletons and catastrophically impact the 3D estimation result in step 3. The traditional one-directional pipeline approach will not improve the parameters of step 1 as each module works independently while having an end-to-end training mechanism allows the model to keep improving the parameters of each step as a result.

However, there is still one bottleneck when we carry out this design. The matching algorithm in step 2 makes the pipeline discontinuous, i.e., it is not a smooth function in which we can back-propagate the changes in parameters freely. However, we can reconcile this with inspiration from Capsule Networks [49, 97]. In CapsNet, the Dynamic Routing step decides how lower layer capsules are fed to their immediate upper layer, either by agreement or expectation-maximization (EM) clustering. In our work, the matching algorithm in step 2 acts in a very similar fashion to the Dynamic Routing. It also decides the feed-forward paths in which information flows from step 1 to step 3, i.e., we apply our matching algorithm to dynamically route/match the poses. This justification and analogy makes our end-to-end approach highly appropriate and is the central theme of our paper.

As one may appreciate, in this end-to-end training mechanism, the dynamic matching step plays a pivotal role. Hence it is vital that we also improve upon the existing works in this area. To this end, we additionally propose a novel matching algorithm which can match multiple 2D poses from multiple views efficiently. The algorithm is robust and can handle situations where there is incomplete and false 2D detection.

In summary, the main contributions of our work are stated below:

- We propose a novel end-to-end training scheme for multi-view multi-person 3D

pose estimation. Different from training independent modules separately, our model back-propagates the gradients from the last 3D estimation step to the first 2D detection step, so as to significantly improve the efficiency, robustness and accuracy on 3D pose estimation.

- We propose a multi-view 2D human pose dynamic matching algorithm. This could dynamically match the corresponding 2D poses detected in multiple views for each person involved. The approach does not require the exact number of people in the scene and can handle cases where false detection and severe occlusions exist.

- Experiments on the Shelf and Campus datasets demonstrate that our proposed model outperforms the state-of-the-art methods with respect to both efficiency and accuracy.

3.2 Related Work

In this section, we review the literature related to the techniques of this paper.

3.2.1 Single-view 2D Pose Estimation

Single person pose estimation predicts 2D keypoints of the human body in one RGB image. Many existing deep learning-based methods have achieved amazing results [82, 52, 10] since DeepPose [110] was proposed, which was the first method to use deep neural networks for pose estimation.

For multi-person 2D pose estimation, current state-of-the-art solutions can be divided into two categories. The first category is called the "top-down methods" [15, 30, 55]. It uses an object detection method to detect all the people in the image and sends them separately to a single 2D pose detector to obtain their corresponding 2D poses. In [55], the authors constructed a fully connected graph from a set of

detected joint candidates of each person in an image and resolved the joint-to-person association and outlier detection by using integer linear programming. [30] proposed a framework with three components for pose estimation which can extract a high-quality single person region from an inaccurate bounding box. In [15], a two-part network structure was proposed where GlobalNet localizes the "simple" keypoints and the RefineNet deals with the "hard" keypoints. The second category, "bottom-up methods", jointly labels part detection candidates and associates them with individuals by a matching algorithm [88, 9, 53]. The authors in [9] mapped the relationship between keypoints into part affinity fields (PAFs), then clustered detected keypoints into different 3D human poses. [88] interpreted the problem of distinguishing different people in an image as an Integer Linear Programming problem and partitioned part detection candidates into identity clusters. On the basis of [88], the authors in [53] used a stronger part detectors based on ResNet [45] and image-dependent pairwise scores, vastly improving the run time by using an incremental optimization approach.

In our work, we choose the "top-down methods" for their higher accuracy. We adopt the Cascaded Pyramid Network (CPN) [15] as the 2D pose estimator backbone.

3.2.2 Multi-view 3D Pose Estimation

Instead of estimating with a single image, multi-view 3D pose estimation methods require image inputs from multiple views, which are believed to obtain better 3D pose estimation than using a monocular camera. Most previous efforts had focused on single person estimation [58, 101]. Traditional methods [2, 5, 8] used 2D pose estimation captured by calibrated cameras to predict 3D poses by point triangulation or 3DPS. Recent works have begun to adopt deep neural networks in this area and have delivered significant achievements. For example, in [57], a volumetric

triangulation approach was proposed to project the feature maps produced by 2D pose estimators into 3D volumes, which were then used to predict 3D poses. There are also self-supervised approaches that predict 3D poses separately in different camera views and minimize the distance between pairwise 3D poses after rotating to the same view [64, 95, 14].

As for multi-view multi-person 3D pose estimation, 3DPS is the most widely used approach [3, 4, 59]. It predicts 3D keypoints or 3D body parts by exploring an ample state space and the candidates in the state space are generated by the grid sampling. With the 2D priors given by the 2D detector, the 3D pose can be generated through the maximum likelihood estimation. Recent work [27] has proposed a model to combine person re-identification (re-id) [128, 129] and epipolar geometry to match the pose, followed by the prediction of 3D poses using 3DPS. The shortcoming of this approach is that the speed of the person re-id model is relatively slow, which causes efficiency problems. On the contrary, our approach is efficient on multi-view multi-person 3D pose estimation, which benefits from our novel matching algorithm.

3.2.3 Dynamic Routing

Dynamic routing is a technique used in CapsNet whereby a capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity such as an object or an object part. Through dynamic routing, lower layer capsules are "selectively" fed into their immediate upper layer. There are two routing algorithms. The first one, proposed in [97], is agreement-based, which calculates the output of a capsule with several consecutive functions so as to determine whether an upper layer capsule "agrees" with this output. The other one, proposed in [49], called EM routing, clusters the capsules in the lower layer and sends the weighted (determined by clustering results) inputs to the capsules in

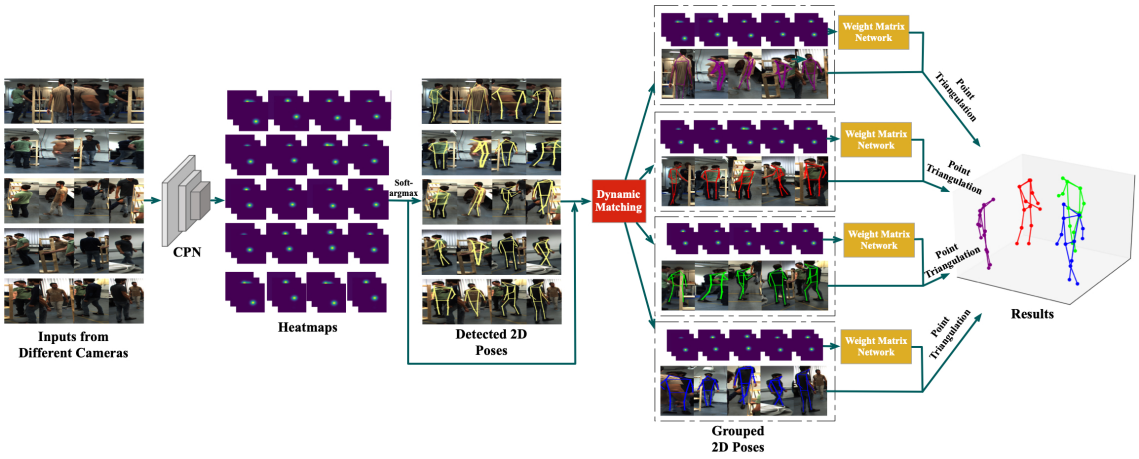

Figure 3.1 : The framework of our proposed model. First, the images I are input into the 2D human keypoints detector backbone, which is based on CPN [15], to get the heatmaps h. Next, we apply soft-argmax on h to get the corresponding 2D human poses y. Then, we feed both h and y into the dynamic matching module which groups them by identities and automatically determines the number of groups. After that, the heatmaps are sent into a network to get the weight matrices. Last, each cluster is sent to a weight-sharing 3D pose estimator to get the final results Y.

the upper layer. Inspired by the ideas behind the above-named dynamic routing, we design our end-to-end model with the dynamic matching algorithm, which can back-propagate the gradients in a similar way.

3.3 Method

In this section, we demonstrate our proposed end-to-end 3D pose estimation model in detail. The scenario assumes there are synchronized video streams from multiple cameras with known parameters, and all cameras capture the same scene with one or more people in it from different views. The goal is to estimate the 3D positions of the keypoints of these people. Note that the exact number of people in the scene is not required.

The inputs of the model are cropped 2D human images from all cameras in the same frame. The images, denoted by I, are cropped by using bounding boxes from either available off-the-shelf 2D human bounding box detectors or ground truths. $I = \{I_n^c | c = 1, 2, \ldots, C, n = 1, 2, \ldots, N_c\}$ where I_n^c is the nth image in the cth view, C is the number of views and N_c is the number of detected bounding boxes in the cth view. The outputs, denoted by Y, are the 3D keypoints of all detected people in the scene. The overview architecture of our model is illustrated in Fig. 3.1.

In the following text, we will demonstrate the 2D pose estimator backbone, dynamic matching algorithm and 3D pose estimation module respectively.

3.3.1 2D Pose Estimator Backbone

The 2D pose estimator backbone f_p with trainable weights θ_p consists of GlobalNet and RefineNet. The GlobalNet predicts all keypoints while the RefineNet justifies the "hard" keypoints. The backbone outputs the heatmaps:

$$h_n^c = f_p\left(I_n^c; \theta_p\right), c = 1, 2, \ldots, C, n = 1, 2, \ldots, N_c. \tag{3.1}$$

The next step is to estimate the 2D positions. To keep the gradient flow, we use soft-argmax instead of argmax to the heatmaps across spatial axes:

$$g_{n,j}^c = e^{h_{n,j}^c} / \left(\int_{q \in \Omega} e^{h_{n,j}^c(q)}\right), \tag{3.2}$$

where $h_{n,j}^c$ denotes the heatmap of the jth keypoint of the nth detected person in the cth view and Ω denotes the domain of the heatmap. Then the 2D coordinates of the estimated joint $y_{n,j}^c$ is the integration of all locations q in the domain, weighted by their corresponding probabilities (we use y_n^c to denote the 2D coordinates of all keypoints of the nth detected person in the cth view):

$$y_{n,j}^c = \int_{q \in \Omega} q * g_{n,j}^c(q). \tag{3.3}$$

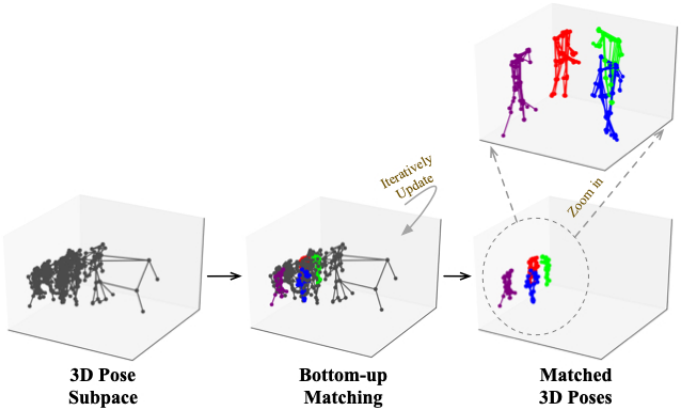

Figure 3.2 : Overview of the our matching algorithm

3.3.2 Dynamic Matching

A matching algorithm is to group 2D poses from different views with people's identities so as to connect the 2D pose detection and 3D pose estimation. It is a challenging task due to several reasons. First of all, there are sizable errors in the estimated 2D poses which can significantly influence the matching accuracy. The second reason is that the number of people in the scene is unknown, which means one cannot cluster these 2D poses to centers like what k-means does. Furthermore, the matching itself is hard to be cycle-consistent. For example, 2D poses y_1^1 and y_1^2 are matched, so do y_1^1 and y_1^3, but y_1^2 and y_1^3 are not matched.

Different from previous methods which compute the matching score for 2D poses, we propose a new matching algorithm that creates a 3D pose subspace first and recursively finds matched 3D poses in this subspace. It resolves both the efficiency and cycle-consistent problems simultaneously. This newly proposed matching algorithm is illustrated in Fig. 3.2.

3D pose subspace construction

To construct the 3D pose subspace, we first enumerate all possible pairs of 2D poses from different views. For each pair of 2D poses, we apply the traditional

point triangulation to generate the corresponding 3D pose. All generated 3D poses compose a 3D pose subspace containing a small quantity of correct 3D poses (i.e., matched 2D poses) and a large quantity of incorrect 3D poses. For each pair of 2D keypoints $y^c_{n,j}$ and $y^d_{m,j}$, $c \neq d$, we can get the coefficient matrices for their corresponding homogeneous 3D vectors:

$$A^c_{n,j} = \begin{bmatrix} y^c_{n,j} \\ 1 \end{bmatrix} \times P_c, \qquad A^d_{m,j} = \begin{bmatrix} y^d_{m,j} \\ 1 \end{bmatrix} \times P_d, \tag{3.4}$$

where P_c and P_d are the projection matrices of cameras c and d respectively. Thus, the 3D point $\tilde{Y}_{(c_n,d_m),j}$ can be obtained by solving the following linear system:

$$\begin{bmatrix} A^c_{n,j} \\ A^d_{m,j} \end{bmatrix} \cdot \begin{bmatrix} \tilde{Y}_{(c_n,d_m),j} \\ 1 \end{bmatrix} = 0. \tag{3.5}$$

We use $\tilde{Y}_{(c_n,d_m)}$ to denote the calculated 3D pose given 2D poses y^c_n and y^d_m. The number of 3D poses constructed is

$$T = \sum_{c=1}^{C} N_c \sum_{d=c+1}^{C} N_d. \tag{3.6}$$

Bottom-up matching

After the construction of 3D pose subspace, we now need to pick out the correct 3D poses. The idea we distinguish the correct 3D poses with incorrect ones is that, the correct 3D poses are almost always calculated by 2D poses belonging to the same person. For example, if a person is captured by four cameras, we will detect four 2D poses which are used to construct six 3D poses, and these 3D poses are almost always very similar to each other, i.e. their distances are very small. Therefore, if the distance between a pair of 3D poses is sufficiently small, their corresponding 2D poses are regarded as a match.

We use the euclidean distance as the measurement between pairwise 3D poses

$\tilde{Y}_{(c_n,d_m)}$ and $\tilde{Y}_{(c'_p,d'_q)}$:

$$E(\tilde{Y}_{(c_n,d_m)}, \tilde{Y}_{(c'_p,d'_q)}) = \|\tilde{Y}_{(c_n,d_m)} - \tilde{Y}_{(c'_p,d'_q)}\|_F, \tag{3.7}$$

where $\|\cdot\|$ is the Frobenius norm. Since we do not need to calculate the distance between 3D poses coming from the same views (i.e. $c = c'$ and $d = d'$), the number of distances calculated is

$$|D| = \sum_{c=1}^{C} \sum_{d=c+1}^{C} (T - N_c N_d) \cdot N_c N_d / 2. \tag{3.8}$$

where D denotes the set of distances between all possible pairwise 3D poses and $|\cdot|$ here is the cardinality.

In order to efficiently obtain all matches, we propose a bottom-up matching algorithm. Suppose the matching result is stored in a set $S = \{s_k | k = 1, 2, \dots\}$ where s_k is a subset which contains the indices of 2D poses belonging to the same person. We initialize S as an empty set and update it by iterations. In each iteration, we first find the minimal distance in D, denoted by $D_{\min}$ which relates to two 3D poses generated by four 2D poses (three if one of them is shared by both pairs), say $y_{n_1}^{c_1}$, $y_{n_2}^{c_2}$, $y_{m_1}^{d_1}$ and $y_{m_2}^{d_2}$, and their corresponding indices can be denoted by a set of view-image pairs $V = \{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\}$. Next, we find a subset s_k^* in S which contains any of the indices in V. If no subset is found, we add an empty set $s_k^* = \{\}$ into S. This finding process is referred as $F(S, V)$. Then we update s_k^* by $s_k^* = s_k^* \cup V$. Note that an index will be dropped if s_k^* has already contained another index from the same view. After the update, $D_{\min}$ will be removed from D. We repeat the above steps until $D_{\min} > \rho$ where ρ is a predefined threshold. The complete bottom-up matching algorithm is presented in Algorithm 1.

Through the matching algorithm we can get the resultant $S = \{s_1, s_2, \dots, s_K\}$ where K is the estimated number of people in the scene. It is determined automatically by the algorithm. According to the indices in s_k we can select the 2D poses

Algorithm 1 Bottom-up matching algorithm

Input: D, ρ

Output: S

1: $InitializeS \leftarrow \emptyset$

2: $D_{\min} \leftarrow \min(D)$

3: **while** $D_{\min} < \rho$ **do**

4: $\quad \{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\} \leftarrow D_{\min}$

5: $\quad V \leftarrow \{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\}$

6: $\quad s_k^* \leftarrow F(S, V) \cup V$

7: $\quad D \leftarrow D \setminus D_{\min}$

8: $\quad D_{\min} \leftarrow \min(D)$

9: **end while**

and heatmaps of the kth person and group them together:

$$y^{(k)}, h^{(k)} = G\left(y, h, s_k\right), k \in [1, K], \tag{3.9}$$

where y and h are the 2D poses and heatmaps for all people from all views, and function $G(\cdot)$ does the operations of both selection and grouping. Each group of 2D poses and heatmaps will be sent to the subsequent module for 3D pose estimation.

This dynamic matching module plays a similar role as the dynamic routing (especially the EM routing) in CapsNet. The difference between them is that the dynamic routing integrates the features from lower capsules by using weighted summation, while our dynamic matching clusters the 2D poses and corresponding heatmaps without any value changes.

Note that the proposed dynamic matching requires at least three views of the scene, which can be inferred form Eq. (3.8). When there are only two views, $|D|$ in Eq. (3.8) becomes 0, which invalidates the whole matching algorithm. Therefore,

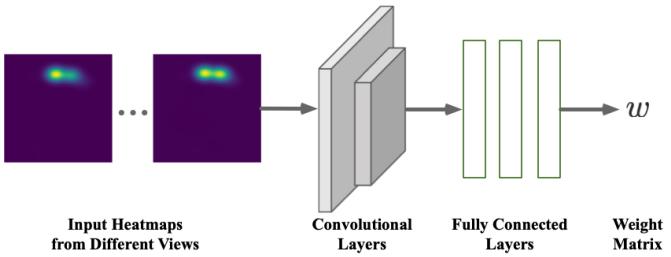

Figure 3.3 : The structure of the weight matrix network

for this special case of two views, we use auxiliary approaches such as the above mentioned person re-id and epipolar geometry.

3.3.3 3D Pose Estimation

Given the grouped 2D poses and heatmaps of each person, we can reconstruct their 3D poses in several ways. The point triangulation described previously is one of them. However, we are now using the 2D keypoints from all views instead of a pair of views, and the corresponding linear system becomes:

$$A_j^{(k)} \cdot \begin{bmatrix} Y_j^{(k)} \\ 1 \end{bmatrix} = 0, \tag{3.10}$$

where $A_j^{(k)}$ is a matrix concatenating the homogeneous 3D vectors of all views for the $j$the keypoint of the kth person.

The point triangulation is an efficient 3D pose estimation algorithm with strong theoretical supports but often produces imprecise 3D poses if there are erroneous detection of 2D poses. The reason is that the coordinates of different keypoints are computed separately. This phenomenon can occur quite frequently at the beginning of training when the 2D pose detection module has not been trained well enough, which in turn affects the improvements of the 2D detection.

To deal with the inaccuracy, inspired by [57], we add a learnable module f_w illustrated in Fig. 3.3 before the point triangulation, which accepts the heatmaps

as inputs:

$$w_j^{(k)} = f_w\left(h_j^{(k)}; \theta_w\right). \tag{3.11}$$

The output $w_j^{(k)}$ is a weight matrix which is in the same size of $A_j^{(k)}$. We add it to Eq. (3.10) and have

$$\left(w_j^{(k)} \circ A_j^{(k)}\right) \cdot \begin{bmatrix} Y_j^{(k)} \\ 1 \end{bmatrix} = 0, \tag{3.12}$$

The original module in [57] predicts a scalar weight for each view denoting how important the keypoints of a view will be. However, scalar weights cannot reflect the details of importance. For example, if a detected keypoint is inaccurate on the horizontal axis but very accurate on the vertical axis, scalar weights have to balance their importance and there will be no difference of importance if we switch the accuracy for both axes. Therefore, we propose to use a weight matrix instead of a scalar weight to better learn the importance so that the accuracy of point triangulation can be further improved.

3.3.4 Loss Function

Our loss function contains two parts, the 2D reprojection loss and the 3D mean square error (MSE) loss. The reason we add the 2D reprojection loss is that, if we only use the 3D MSE loss, there would be infinite points that have the same loss value but target at the 3D ground truth in different directions. The 2D reprojection loss can indicate the correct direction by constraining projected 2D poses from different views.

The 3D MSE loss between the estimated 3D pose and 3D ground truth is defined as:

$$L_{\text{mse}}^{\text{3d}} = \sum_{k=1}^{K} \frac{1}{|Y^{(k)}|} \|Y^{(k)} - Y_{gt}^{(k)}\|_F^2. \tag{3.13}$$

The 2D reprojection loss between the reprojected 2D pose from the computed 3D pose and the detected 2D pose from backbone is defined as:

$$L_{\text{repj}}^{\text{2d}} = \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{1}{|y_c^{(k)}|} \|\tilde{y}_c^{(k)} - y_c^{(k)}\|_F^2, \tag{3.14}$$

where

$$\tilde{y}_c^{(k)} = \left[p_1 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} / p_3 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix}, p_2 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} / p_3 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} \right], \tag{3.15}$$

and

$$P_c = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T. \tag{3.16}$$

Thus, the total loss of our model is defined as:

$$L = L_{\text{mse}}^{\text{3d}} + \alpha L_{\text{repj}}^{\text{2d}}, \tag{3.17}$$

where α is a weight coefficient.

3.4 Experiments

3.4.1 Datasets

We conduct experiments on two standard datasets for multi-view multi-person 3D human pose estimation.

Shelf [3]: The Shelf dataset is one of the public 3D multi-person human pose datasets in multi-view setting. It consists of 3200 frames from 5 synchronized cameras along with the 2D pose annotations and 3D pose ground truth derived by pose triangulation. There are 4 human subjects interacting with each other in a small room. All 3200 frames are split into an evaluation set (frame 300-600) and a training set (other frames).

Campus [3]: The Campus dataset contains three human subjects interacting with each other in an outdoor environment. The scene is captured by three calibrated cameras. The dataset consists of 2000 frames and is divided into an evaluation set (frame 350-470, frame 650-750) and a training set (other frames).

For the evaluation protocol, we use the percentage of correctly estimated parts (PCP@0.5) to measure the model performance, which is the most commonly adopted in this area [3, 27].

3.4.2 Implementation Details

As for the data preprocessing, we crop the images with bounding boxes estimated by an off-the-shelf 2D human detector, Yolo [92]. The 2D pose detection backbone is the same as [15] with pretrained weights, which outputs heatmaps and connects to a soft-argmax function to obtained the 2D poses. The dynamic matching module is implemented according to Algorithm 1. The 3D pose estimator consists of two convolutional layers and three fully-connected layers. The weight coefficient α in the loss function is set to 2. We choose the Adam optimizer with a learning rate of 10^{-6} which reduces by a decay factor of 10 in each epoch. The training set and evaluation set are kept the same as described in the datasets.

Table 3.1 : Matching results of different threshold value on the Shelf dataset.

Threshold value	0	20	40	100	1000	10000
Matching number	0	2	**4**	**4**	**4**	5

As for the threshold value, we conducted experiments with its different values on the Shelf dataset and reported the number of matched 3D poses in Table 3.1. The correct matching number is 4. From the table we can see that, the result is correct when the threshold is set between 40 and 1000. The idea behind this choice is simple. Suppose that the distance between most correct pairs are in $[a, b]$, while that of most incorrect pairs are in $[c, d]$. Because c is much larger than b, we can choose any value in $[b, c]$ as the threshold. Here, $a = 0, b = 40, c = 1000, d = 10000$. In our implementation, we choose 40 as the threshold for both datasets.

3.4.3 Ablation Study

Our first experiment is to verify the effectiveness of different settings for our model through the ablation study on the Shelf dataset.

Table 3.2 : The PCP@0.5 performance of the alternative multi-step model and our end-to-end model on the Shelf dataset. They are using the same 2D pose detection backbone, matching algorithm, 3D pose estimator and loss function.

	Actor 1	Actor 2	Actor 3	Average
Multi-step	98.12	95.16	96.77	96.67
End-to-end (ours)	**98.75**	**96.22**	**97.20**	**97.39**

End-to-end vs Multi-step Architecture

Our model is end-to-end and can predict the 3D poses from 2D human images as a whole. An alternative is to divide the model into three consecutive steps which deal with the 2D pose detection, matching and 3D pose estimation separately. We compare these two architectures and the results are presented in Table 3.2.

From the table, we can see that the performance of our end-to-end model is better than the multi-step model for all three people in the scene. The average improvement is 0.72. This demonstrates that the end-to-end model is more capable of learning the features of human poses which refines the 2D pose detection with gradients flowing back from the overall loss function.

Matching Method

Given the 2D poses obtained from the 2D detection module, we propose a novel matching algorithm to group the 2D poses and heatmaps by identities. There are two existing matching methods in the literature, the person re-id and epipolar geometry.

Table 3.3 : Comparison of matching methods including the person re-id, epipolar geometry and our algorithm on the Shelf dataset over the PCP@0.5 and time cost. All three methods use the same 2D pose detector and 3D pose estimator.

	Actor 1	Actor 2	Actor 3	Average	Time (s)
Person re-id	97.62	93.72	95.69	95.68	6.73
Epipolar geometry	97.28	91.76	91.27	93.44	**0.64**
Our method	**98.75**	**96.22**	**97.20**	**97.39**	0.96

The former finds matches by using the re-id appearance matrix as confidence scores, while the latter uses epipolar geometry affinity matrix as the confidence scores. The comparison between these three matching methods is shown in Table 3.3.

The results show that our matching method achieves the best performance among the three, with average improvements of 1.71 and 3.95. The time cost of person re-id is the highest while that of epipolar geometry is the lowest. Our matching method is slightly slower than epipolar geometry, but still much faster than person re-id. This experiment demonstrates that our matching algorithm is robust and efficient. The reason is that both person re-id and epipolar geometry use 2D information, thus there may be cases where the poses of different people result in a larger confidence score than those of the same person because of the angle of camera views or imprecise 2D detection. On the contrary, our method finds the matches in the 3D pose subspace directly, which leverages the information inequality between the 2D and 3D spaces and makes our method more robust and insensitive to imprecise or even incorrect 2D poses.

Table 3.4 : Performance of our 3D pose reconstruction method compared with the point triangulation and learnable triangulation on the Shelf dataset. They are implemented with the same 2D pose detection backbone and dynamic matching.

	Actor 1	Actor 2	Actor 3	Average
Point triangulation	98.05	91.17	92.78	94.00
Learnable triangulation	98.64	95.83	96.91	97.13
Our method	**98.75**	**96.22**	**97.20**	**97.39**

3D Pose Estimation Method

As described in the method section, we use the point triangulation with a learnable weight matrix to estimate 3D poses. Alternatives include the sole point triangulation or the original learnable triangulation network [57]. We compare these two methods with ours and the result is presented in Table 3.4.

We can see from the table that our method outperforms the other two methods by 3.39 and 0.26 respectively in average. This demonstrates that (1) the 3D poses estimated by point triangulation is not accurate enough, (2) adding learnable scalar weights can significant improve the performance and (3) using a learnable weight matrix instead of the scalar weights can further improve the model's robustness.

3.4.4 Comparison with Previous Works

We compare our model with existing state-of-the-art models for multi-view multi-person 3D pose estimation on both datasets. The models compared are:

- Belagiannis et al. [3], the first one applying the 3DPS to 3D pose estimation for multiple humans.
- Belagiannis et al. [4], an improved version of their previous work.

Table 3.5 : Comparison of multi-view multi-person 3D pose estimation models on the Shelf and Campus datasets under PCP@0.5. All results are obtained from the original papers except for the (*) which only provides the average performance (in the parentheses) and its results on body parts presented here are from our own experiments using the authors' published code.

Shelf dataset		Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts	Average
Belagiannis et al. [3]	Actor 1	89.30	90.20	72.16	60.59	37.12	70.61	66.05	
	Actor 2	72.10	92.80	80.11	44.20	46.30	71.80	64.97	71.39
	Actor 3	94.66	96.35	91.00	89.00	45.80	94.50	83.16	
Belagiannis et al. [4]	Actor 1	96.29	100.00	82.24	66.67	43.17	86.07	75.26	
	Actor 2	78.95	100.00	82.58	47.37	50.00	78.95	69.67	77.51
	Actor 3	98.00	100.00	93.15	92.30	56.50	97.00	87.59	
Ershadi-Nasab et al. [28]	Actor 1	98.27	97.34	92.57	83.33	95.94	96.83	93.29	
	Actor 2	63.05	94.61	78.33	33.38	95.30	93.45	75.85	87.99
	Actor 3	98.15	94.12	94.43	89.82	97.41	96.34	94.83	
Dong et al. [27]*	Actor 1	88.17	100.00	99.82	99.28	99.82	100.00	98.60	
	Actor 2	97.30	100.00	98.65	71.62	100.00	100.00	93.78	96.76 (96.90)
	Actor 3	94.41	100.00	95.96	96.27	100.00	100.00	**97.89**	
Our model	Actor 1	88.89	100.00	99.82	99.46	100.00	100.00	**98.75**	
	Actor 2	100.00	100.00	100.00	81.08	100.00	100.00	**96.22**	**97.39**
	Actor 3	90.06	100.00	95.65	95.96	95.96	99.38	97.20	
Campus dataset		Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts	Average
Belagiannis et al. [3]	Actor 1	93.62	49.94	82.85	77.80	86.23	91.39	82.01	
	Actor 2	97.40	41.13	90.36	39.65	73.87	89.02	72.43	75.79
	Actor 3	81.26	69.67	77.58	61.84	83.44	70.27	73.72	
Belagiannis et al. [4]	Actor 1	96.55	93.10	96.55	86.21	93.10	96.55	93.45	
	Actor 2	98.24	48.82	97.35	42.94	75.00	89.41	75.65	84.49
	Actor 3	93.20	85.44	89.81	74.76	91.75	76.21	84.37	
Ershadi-Nasab et al. [28]	Actor 1	97.31	94.16	96.83	87.48	93.67	97.27	94.18	
	Actor 2	98.73	95.41	94.12	78.98	98.94	95.34	92.89	90.56
	Actor 3	95.36	84.37	93.16	70.34	88.36	81.38	84.62	
Dong et al. [27]*	Actor 1	100.00	100.00	97.96	89.80	100.00	100.00	97.55	
	Actor 2	97.88	100.00	100.00	67.72	100.00	100.00	93.33	95.85 (96.30)
	Actor 3	99.28	99.28	98.91	89.86	97.46	97.83	96.67	
Our model	Actor 1	100.00	100.00	98.98	90.82	100.00	100.00	**97.96**	
	Actor 2	99.47	100.00	100.00	74.34	100.00	100.00	**94.81**	**96.71**
	Actor 3	100.00	100.00	99.64	90.58	97.10	97.46	**97.39**	

- Ershadi-Nasab et al. [28], an extension of the 3DPS.
- Dong et al. [27], which uses person re-id and geometry methods to match 2D

poses.

For the Campus dataset, since the number of views is insufficient to generate enough 3D pose candidates, we use person re-id and epipolar geometry as auxiliaries in our matching algorithm. The comparison results are shown in Table 3.5.

On both datasets our model surpasses the state-of-the-art methods in almost all cases. The average performance of our model is 97.39 and 96.71 respectively with improvements of 0.63 and 0.86 comparing with the second best model (0.49 and 0.41 improvements if compared with the results from their paper). It is noteworthy that, the performance of existing models on the lower arms of Actor 2 in Shelf dataset is quite low, while ours achieves 81.08 with a huge improvement of 9.46. We notice that there exists a large quantity of occlusions in this case, which means our model can better handle occlusions than others in a multi-person setting.

3.5 Conclusion

In this paper, we have proposed a novel end-to-end dynamic matching network for multi-view multi-person 3D pose estimation. Different from previous studies, the end-to-end scheme of our work enables the gradients to flow back from the 3D pose estimation module to the 2D pose detection backbone. A bottom-up dynamic matching algorithm is proposed to group the 2D poses and heatmaps by identities so as to connect the 2D pose detector and the 3D pose estimator. The algorithm is efficient and robust and able to automatically determine the number of people in the scene. The ablation study verified the effectiveness of each part of our model and the experimental results on the Shelf and Campus datasets demonstrate that our proposed model is superior to the state-of-the-art models with respect to accuracy, robustness and efficiency. This paper discovered a useful setting in pose estimation, however, it needs multiple cameras and lots of ground-truth data. In following

chapters, we begin considering how to use less or no ground-truth data and other kind of data to train a more robust network.

Chapter 4

Self-supervised Network for 3D Human Pose Estimation

4.1 Introduction

Learning to estimate 3D body poses has attracted substantial interest. Various applications can be derived from this technology including human-computer interaction, action recognition, and virtual reality. With the great success of deep learning models and more and more sophisticated datasets, researchers [70, 86] applied deep convolutional neural networks(CNNs) to estimate 3D human poses in a monocular camera setting and have achieved great improvements in this area.

However, methods using neural networks to estimate 3D pose from a monocular camera view face some challenges. Firstly, for most typical neural network models [103, 94, 132], massive training data of annotated 3D human poses are needed. These methods rely on the 3D annotated skeletons for depth predictions. The 3D ground-truth labels are captured by the Motion Capture system, which is too expensive. Secondly, there are well-proven mathematical theories on projecting 2D joints into 3D space, using neural networks is an approximation of this projection. There are concerns that overly relying on the 3D ground-truth data may lead to an over-fitting problem. Thirdly, methods that train on monocular camera view cause the depth ambiguity problem. This is because there are multiple different 3D skeletons that can be projected into the same 2D pose at one specific camera view, so the estimated 3D pose from a monocular view may result in a strange configuration.

In this work, we propose a self-supervised training method in a multi-view train-

ing setting to alleviate the above challenges. In Chapter 3, we use a multi-view multi-person setting to train a network. However, it needs lots of ground truth data to train the network. Instead of explicit 3D ground-truth, we use the self-supervised method, which only requires 2D pose estimations as inputs that can be usually generated by a 2D pose estimator. The multi-view setting not only overcomes the depth ambiguity problem but also resolves the situations of incomplete or false 2D detection by utilizing the information from other views. Typically, a multi-view setting has more advantages in training a model because of the information of different views and a monocular setting has wider application scenarios. We apply the multi-view setting only during the training and apply the monocular setting during the testing, so our method combines the benefits of both paradigms.

The structure of our method mixes the outputs of two branches of the neural networks. The first branch takes a single image (from one camera view) as input and generates the 3D pose in a 3D space, and the second branch inputs three images of different camera views and outputs the estimated 3D pose. Our method allows for the projection of all estimated 3D poses from two branches to any camera view.

In practice, our approach consists of two stages. The first stage estimates the 2D human pose using an off-the-shelf 2D pose estimator. After that, the second stage lifts these 2D estimations into the 3D space. Combining the geometry information of the cameras, a 3D pose from the second branch with the rotation matrix from the view of the first branch results in a rotated 3D pose in the first camera coordinate system. In other words, all 3D poses in the real-world coordinate system should be exactly the same and can be projected back into their own 3D camera coordinate systems. Besides, the 3D poses are re-projected to each 2D camera view through camera matrices, which enables the definition of a re-projection loss for each 3D-2D projection.

We evaluate our approach on two largest benchmark 3D datasets: MPI-INF-3DHP [76] and Human3.6M [54]. The experiments demonstrate that our model outperforms the state-of-the-art methods. In summary, the main contributions are summarized as below:

- We propose a two-branch self-supervised approach in a multi-view training setting to train the 2D-3D neural network without the 3D ground-truth labels. The whole model only relies on the geometry information to build supervision signals.

- We propose a cycle-view training scheme, which is effective in exploiting multi-view consistency and constraining the 3D estimations in the training stage. The method overcomes the depth ambiguity problem and can handle the situations of incomplete or false 2D detection by utilizing the information from other views. Moreover, we make use of the 2D confidence from different cameras to solve the occlusion problem.

- We propose the volumetric 3DPS network based on the voxel-based method and 3DPS. It explores an ample state space and the volumetric cubes in the state space are generated by the grid sampling. The cubes are then sent to the voxel-based neural network to get a 3D pose.

- Our evaluations of the Human3.6M and MPI-INF-3DHP datasets demonstrate that our proposed model achieves state-of-the-art results compared with recent self-supervised methods.

4.2 Background and Related Work

In this section, We review existing work for human pose estimation related to this paper.

4.2.1 2D Human Pose Estimation

2D human pose estimation estimates the 2D joints of a human body in one camera view. Recently many researchers used convolution neural networks and have achieved great improvements. The methods can be generally divided into two classes. The first class is called the " top-down method " [15, 30]. These methods first used a detection model to find all possible humans in the view and estimated each 2D pose of them separately. The second class is called the " bottom-up method " [9]. They first detected all possible body part candidates, then associated them with individuals through a matching algorithm. Due to the great performance of these 2D human poses estimation methods, they are commonly used as a pre-used 2D pose estimator in many 3D estimation works. In our work, we choose the Cascaded Pyramid Network [15] as our 2D pose estimator.

4.2.2 Monocular 3D Human Pose Estimation

Most state-of-the-art methods in this area use images that are annotated with 3D ground-truth skeletons to train the deep neural networks. These methods can be generally divided into two categories. The first category of methods [107, 85, 41, 104] directly estimates the depth of images through the convolutional neural networks. The second class of methods [75, 31, 81, 12] is a two-stage pipeline. They first estimate the 2D key-points of the objects through an off-the-shelf 2D keypoint estimator and then lift them into 3D key-points. Both of them require accurate annotations for training. Among these works, many neural networks were designed to resolve the 2D-3D joint lifting. [75] used a simple neural network consisting of two linear layers and achieved surprisingly well results. [81] encodes pairwise distances of 2D and 3D body joints into two Euclidean Distance Matrices (EDMs) first, and then regresses the 3D EDM through a neural network. Some work [132, 86, 90] exploited temporal information between consecutive frames to alleviate the need of

3D annotations and produce more robust results. Several works [124, 36] added adversarial losses in their model to improve the performance. Our model is designed in the same way as the two-stage pipeline.

4.2.3 Multi-view 3D Human Pose Estimation

Multi-view 3D human pose estimation methods receive inputs from several different views, which attracted more attention recently due to better performance than using a monocular camera view. Early works [3] used 2D estimations from several calibrated camera views to predict 3D poses through the point triangulation or the 3D pictorial structures model (3DPS). Recently, researchers [27, 85, 57, 51] have begun to adopt novel convolutional neural networks in this area to improve the robustness of the framework and made significant achievements. For example, the authors of [57] presented a network that learns the triangulation process during the training and predicts the 3D pose. These methods use multi-view settings at both training and testing stages and still require 3D annotations. Our proposed model only uses a multi-view setting at the training stage, and back to a monocular setting during testing.

4.2.4 Self-supervised Learning

Recently, self-supervised (weakly-supervised) methods have attracted much attention because they do not require the paired 2D-3D annotations and only use weak supervision or no supervision. The authors of [13] introduced a cycle consistency loss computed by lifting the randomly projected 2D pose to the 3D pose and inversing the previously defined random projection. The authors of [76] proposed an encoder-decoder model that can compute images from one camera view to another. The network could learn a geometry-based human pose representation, and maps pose from 2D to 3D space. In [108], the authors proposed an integrated approach that integrates a 3D pose model trained with probabilistic knowledge of 3D human

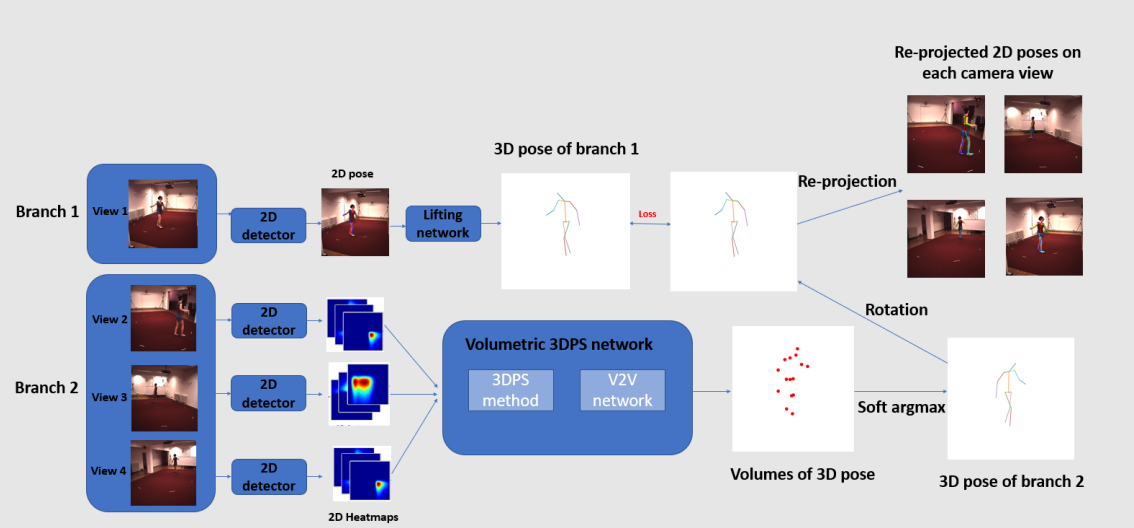

Figure 4.1 : The architecture of our proposed model. The whole network consists of two branches. The first branch inputs a single image (from one camera view) and generates a 3D pose in the 3D space, and the second branch inputs the other three images of camera views and outputs the estimated 3D pose. The second 3D pose is rotated to the first camera view for multi-view consistency loss that enforces the 3D poses estimated from different views to be an identical skeleton up to a geometry transform. Besides, the predicted 3D pose is re-projected to each camera view to get an additional reprojection error.

pose into a multi-stage 2D CNN architecture, which could refine both 2D and 3D predictions iteratively. In order to reduce dependence on the 3D annotations, the theory of multi-view geometry and camera projection was proposed and then many works began to explore geometry-driven methods. In [64], a self-supervised learning method was proposed for 3D human pose estimation, via training the network base on the 3D poses through the epipolar geometry method. Re-projection loss is a widely used technique in multi-view geometry-driven methods. It helps constrain the multi-view information during the training. The approach in [83] extracts the 3D poses with only 2D pose annotations in unconstrained images. The authors used a re-projection loss and a designed canonicalization function as well. Under such a design the network could factor in the effects of viewpoint changes and object deformations. Adversarial loss is also a commonly used technique. It forces the 3D poses estimated by the network to be close to the shape of the real human pose by applying a 3D pose discriminator. The authors of [114] proposed a self-supervised method that uses adversarial supervision instead of 3D ground-truth for 3D human pose estimation. The authors of [13] presented an unsupervised learning approach using the geometric self-consistency method. They randomly transformed the 3D poses to a different direction in 3D space, and the re-projected 2D poses were constrained by the 2D pose discriminator. Even though the adversarial loss does not need the paired 2D-3D annotations during training, it still requires the unpaired 3D pose annotations to pre-train the 2D/3D pose discriminator.

4.3 Proposed Methodology

In this section, we will present our proposed self-supervised model in detail. Our method follows a two-stage pipeline. First, we apply an off-the-shelf 2D pose estimation network to predict both 2D poses and heatmaps from four input frames. Then we lift these detections with the confidence of all 2D key-points into 3D. Fig.

4.1 shows the whole network structure with four camera views.

We assume there are four synchronized camera views with camera projection matrices, all cameras capture the same person and the same scene. First, we use an off-the-shelf 2D pose estimator to get 2D predictions. For each frame, we crop the images with the bounding boxes detected by an available detector, the cropped image of the first camera view is denoted by I, and the cropped images of the other three camera views are denoted by $\tilde{I} = \{\tilde{I}^c | c = 1, 2, 3\}$. The cropped images are then fed into the 2D pose estimator. The estimator backbone is denoted by f with weights parameters θ. The estimator consists of two parts, the GlobalNet predicts the pose roughly while the RefineNet refines the " hard " joints.

There are two sub-branches during the training. In the first branch, We denote $X \in \mathbb{R}^{N\times 2}$ as N detected 2D joints, then the 3D lifting network predicts the 3D poses $Y \in \mathbb{R}^{N\times 3}$. Another branch takes inputs of three camera views and outputs the 2D heatmaps $\tilde{H} = \{\tilde{H}^c | c = 1, 2, 3\}$ for each view. Then the volumetric 3DPS network accepts input of the heatmaps and outputs the 3D poses $\tilde{Y} = \{\tilde{Y}^c \in \mathbb{R}^{N\times 3} | c = 1, 2, 3\}$. We estimated the heatmaps through the 2D pose estimator:

$$H = f\left(I; \theta\right), \tag{4.1}$$

$$\tilde{H}^c = f\left(\tilde{I}^c; \theta\right), c = 1, 2, 3, \tag{4.2}$$

the 2D human pose X used for the first branch is estimated through H. A widely used way for this job is argmax algorithm. The algorithm compute the max value point of each heatmap to get the highest probability joint. However, this algorithm will cut the gradient flow so we are not able to train the 2D network. To solve this problem, instead of argmax operation, we apply soft-argmax to the heatmaps:

$$G_k = e^{H_k} / \left(\int_{i\in\Omega} e^{H_k(i)}\right), \tag{4.3}$$

where H_k denotes the heatmap of the kth body joint of the captured person of camera one and Ω denotes the domain of the heatmap. The location i weighted by

its corresponding probability $G_k(i)$ is a weighted coordinated of this location, add all of them up we get the 2D coordinates of the predicted keypoint:

$$X_k = \int_{i \in \Omega} i * G_k(i). \tag{4.4}$$

Following the protocol with previous works, we estimate zero-centered 3D poses where the values of Y and $\tilde{Y}$ are the 3D positions relative to the fixed root joint. The predicted 3D pose is at its own pose coordinate system during training, we then rotate $\tilde{Y}$ to the first camera coordinate system to get the 3D loss, and get the reprojection loss by re-projecting the global 3D pose into the 2D pose of each view.

4.3.1 Lifting Network

The architecture of the lifting network w^v is designed with an inspiration by [75]. The goal of the lifting network is to estimate body joint locations in the 3D space given only a 2D input. The network is based on batch normalization, dropout, and Rectified Linear Units, as well as residual connections. The input layer takes the coordinates of N(in our case, 17) human key-points and applies a fully connected layer with 1024 output channels. It is then followed by four blocks with residual connections. Each block consists of two layers. Each layer is followed by batch normalization, rectified linear units, and dropout. Final features output by the last residual block is fed into a linear layer to get 3D poses Y:

$$Y = w^v\left(X\right). \tag{4.5}$$

4.3.2 Volumetric 3DPS Network

3D Pictorial Structures model (3DPS) is a commonly used method for multi-view 3D pose prediction [3, 4, 59]. It explores an ample state space and generates 3D body part candidates through grid sampling. Then the method predicts the 2D pose

through a 2D detector, with the 2D priors and a maximum likelihood estimation, the 3D pose is generated. However, the traditional 3DPS method is not precise enough for our work. To increase the accuracy and robustness of our model, we apply the volumetric approach.

Instead of sampling the 3D joint, we sample the 3D cube around the person joint through 3DPS. Through the 2D backbone, we already have the heatmaps H_c^k, then we apply the 3DPS method to get the volumetric cubes:

$$V_c^k = T\left(H_c^k\right), \tag{4.6}$$

where V_c^k is the generated volumetric cube of kth joint heatmap of cth camera view, and T denotes the 3DPS method. Here, we set the size of the cube to $32 \times 32 \times 32$. Adding up each view's cube data:

$$V_k^{sum} = \sum_c V_c^k, \tag{4.7}$$

we then fed the cubes into the learnable volumetric convolutional neural network p^q. Its architecture is similar to voxel-to-voxel network [80]:

$$H^{3D} = p^q\left(V^{sum}\right), \tag{4.8}$$

where q denotes the weights of the voxel-to-voxel network p and H^{3D} denotes the 3D heatmaps of the predicted 3D human joints. Then we will estimate the 3D positions. we apply soft-argmax to the heatmaps:

$$G_k^{3D} = e^{H_k^{3D}} / \left(\int_{i\in\Omega} e^{H_k^{3D}(i)}\right), \tag{4.9}$$

where H_k^{3D} denotes the heatmap of the kth body joint of the estimated person in the 3D space and Ω denotes the domain. Then we get 3D estimated keypoint Y_k :

$$\tilde{Y}_k = \int_{i\in\Omega} i * G_k^{3D}(i). \tag{4.10}$$

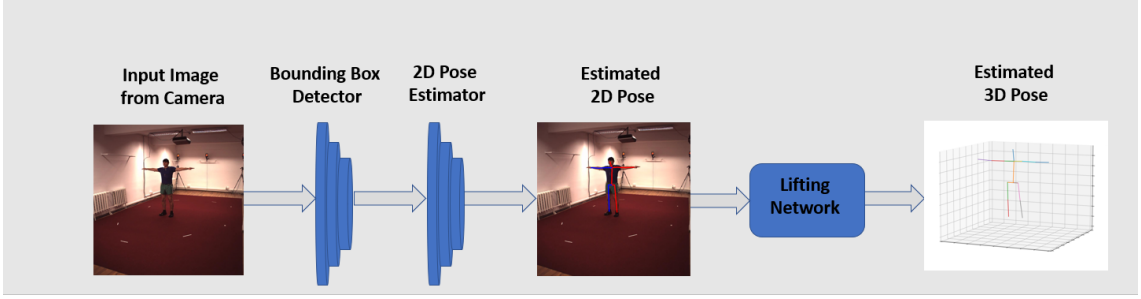

Figure 4.2 : The overall architecture of our method. We input each frame to a Bounding box detector to get cropped images and fed them into 2D pose backbone. We then input the estimated 2D pose to the lifting network to get the final results. Note the multi-view setting is only applied to the training part, the testing stage follows the monocular camera setting.

4.3.3 Cycle-view Training

Usually, an easy way of assuring multi-view consistency is to use the L2 norm between the 3D poses predicted from different views. If the predictions are perfect, the loss should be zero. This is because the same human captured from different camera views should have the identical absolute 3D pose in 3D space. However, for more than two views' situation, the solution does not work well. During the training, the L2 loss of view1 and view2 will conflict with the L2 loss of view2 and view3. This is because we can not obtain the depth information of the camera view, the lifting network learns a 3D pose only suited for one view.

To solve the problem, the key is to find a way that can strongly force the 3D poses generated by different camera views can be transformed into the identical 3D pose. Following this idea, we design a cycle-view training scheme. Our network consists of 2 branches, the first branch is one camera view followed by the lifting network, and the second branch is three camera views followed by the volumetric 3DPS network. We randomly exchange these four views' input images with each

other, so in this way, all possible combinations of the 3D views will be trained. Following this training scheme, we solve the multi-view consistency problem.

4.3.4 Pre-train Scheme

Our model uses a self-supervised way to train the network in a multi-view setting. The model benefits from the information from different views. However, without 3D ground-truth annotations, the 3D poses generated from two branches are difficult to locate in the same absolute 3D positions, which makes our network hard to converge. To solve this problem, we apply a pre-train scheme to help locate the 3D poses. We use another lifting network to predict the center of the 3D pose, and we choose the human pelvis as the center key-point. The center joint predicted network has the same structure as the lifting network and has its own weights. The networks are denoted by u^{γ} and $\tilde{u}^{\tilde{\gamma}}$, where u^{γ} predicts the center joint of camera view one and $\tilde{u}^{\tilde{\gamma}}$ predicts the center joint of a randomly picked view of camera view two, three and four. We predict the center joints in pre-train:

$$Y^{pelvis} = u^{\gamma}\left(X\right), \tag{4.11}$$

$$\tilde{Y}^{pelvis} = \tilde{u}^{\tilde{\gamma}}\left(\tilde{X}\right), \tag{4.12}$$

then we use the predicted center joint to locate the 3D pose and move the 3D pose to the predicted position during the training.

4.4 Experiments

4.4.1 Datasets

We conduct experiments on two available large datasets Human3.6M and MPI-INF-3DHP. Human3.6M (H3.6M) [54] is currently one of the largest 3D human pose benchmarks and is widely used in tasks of both monocular and multi-view settings. It contains 3600 images with 11 actors, of which actor perform 15 actions such as

Table 4.1 : Comparison of two training schemes: constant-view training and cycle-view training. The results are for the H3.6M dataset over two protocols. All schemes are using the same 2D pose estimator backbone, lifting network, 3D pose estimator and loss function. The MPJPE error and PMPJPE error are given in mm.

	MPJPE	PMPJPE
constant training	61.9	53.4
cycle-view training	**57.3**	**48.1**

walking, smoking, and talking. The images are captured from four cameras, and the cameras are all calibrated. We follow the previous work protocol in our experiments, using S1, S5, S6, S7, and S8 for training, using S9 and S11 for testing. MPI-INF-3DHP (3DHP) [76] is another famous 3D pose dataset. Besides indoor scenes, it also includes many complicated in-the-wild images. We use four chest-height cameras (considering compatibility with the H36M dataset, we use the provided 17 joints) for training and the test-set consists of six sequences for evaluation.

Table 4.2 : Comparison of evaluation on the Human3.6M dataset with different backbones. We present the MPJPE error and PMPJPE error and they are given in mm.

	MPJPE	PMPJPE
Residual Linear	59.7	49.3
Temporal Dilated	**56.1**	**47.2**
Our backbone	57.3	48.1

4.4.2 Metrics

For the H36M dataset, we consider two popular evaluation protocols proposed by previous work. Protocol 1 is the MPJPE, which is an average Euclidean distance between the predicted location of the node and the labels. Protocol 2 is the P-MPJPE. Before the final calculation of MPJPE, the estimated 3D pose is aligned by the rigid transformation of the Procrustes analysis to get the P-MPJPE.

The evaluation metrics for the 3DHP dataset include the above two protocols and a third protocol: the adapted Percentage of Correct Keypoints (PCK). The PCK represents the percentage of joints within 15 cm of the actual measurement.

Table 4.3 : The results of the evaluations of the experiments test the generalization ability of our model. Training scheme one is trained on H3.6M and tested on 3DHP. Training scheme two is trained on 3DHP and tested on H3.6M. The other two are the results of original experiments the training and testing sets are from the same dataset. All schemes are using the same 2D pose estimator backbone, lifting network, 3D pose estimator, and loss function. The MPJPE error and PMPJPE error are given in mm.

	MPJPE	PMPJPE
3DHP	102.7	70.1
scheme 1	133.7	88.6
H3.6M	57.3	48.1
scheme 2	75.7	69.3

4.4.3 Implementation Details

In this section, we will discuss some detailed settings of our experiments. As for the data pre-processing, we input the images of four camera views into an off-the-shelf 2D pose estimator. The 2D pose detection backbone is the same as [15] with pre-trained weights, and it outputs heatmaps and 2D poses. we apply a pre-train scheme to help locate the 3D poses. We use a lifting network to predict the human pelvis. We set the Adam as the optimizer and learning rate to 0.001, then train the network for 100 epochs, and use 0.001 as the learning rate. After that, we train the network with a pre-trained pelvis-detected network. The network is trained for 300 epochs with a learning rate starting from 0.001 and drops by 0.1 every 100 epochs. During the evaluation, following the same protocol as previous work, we only use the first branch of our model to predict the relative 3D poses, the second branch of the model and the pelvis-predicted network are not used during the evaluation.

4.4.4 Ablation Study

Cycle-view training

To solve the consistency problem of multiple views during the training, we propose a cycle-view training scheme. Normally, during the training, the camera views are constant, which will cause a consistency problem. We design a training scheme that randomly exchanges the camera views which could alleviate this problem. We have experimented with two training schemes and the comparison between them is shown in Table 4.1. The results show that the cycle-view training scheme outperforms the constant taring scheme, with average improvements of 4.6 and 5.3. This experiment demonstrates that the proposed training scheme is as effective as designed. This is because the method constrains the multi-view consistency well. Normally multiple views will cause conflicts because the prediction of different views only serves its own performance. Our method can make the best use of the multi-

view information by constraining them and forcing the 3D to be identical to achieve good results.

Table 4.4 : Comparison of evaluation on the Human3.6M dataset. We present the MPJPE error and PMPJPE error for recent weakly/self-supervised methods. The MPJPE error and PMPJPE error are given in mm.

	MPJPE	PMPJPE
Pavlakos et al.[85]	118.4	-
Rhodin et al.[95]	122.6	98.2
Wandt et al.[114]	89.9	65.1
Kocabas et al.[64]	76.6	67.5
Chen et al.[13]	-	68
Kundu et al.[68]	85.8	-
Kolotouros et al.[65]	-	62.0
Wang et al.[118]	63.7	-
Wang et al.[116]	86.4	62.8
Li et al.[71]	59.0	49.7
Wandt et al.[115]	74.3	53.0
Iqbal et al.[56]	67.4	54.5
Ours	**57.3**	**48.1**

Backbone Influence

To demonstrate the effectiveness of our model with different network backbones, we choose two different 2D-3D lifting backbones for the experiments. We choose

Residual Linear [75] and Temporal Dilated [86]. The comparison between them is shown in Table 4.2. From the table, we could see our model achieves state-of-the-art results in both of the three backbones. This indicates that our model is not relying on lifting backbones. Besides, the results of Temporal Dilated outperform the other two backbones. This demonstrates that our model could benefit from a better lifting backbone, but the state-of-the-art performance is not because of the backbone but because of the effectiveness of our model.

Generalization Ability

To demonstrate the generalization ability of our model, we have designed a training scheme using different datasets for training and testing. First We use the H36M dataset for training and use the 3DHP dataset for testing. Then we exchange them, use the 3DHP dataset for training and use the H36M dataset for testing. The two datasets are very different from each other. The 3DHP dataset includes many in-the-wide scenes while the H36M dataset only has indoor scenes. The comparison between them is shown in Table 4.3. The result shows that even though the error is higher than the original training scheme due to the extreme difference between the two datasets, our approach is robust enough to estimate reliable 3D poses on an untrained dataset.

4.4.5 Comparison with Previous Works

In this section, we compare our method with the state-of-the-art methods. Table 4.4 shows the results on the H3.6M dataset. The average performance of our model of two protocols is 57.3 and 48.1, outperforming the second-best model with improvements of 1.6 and 1.7. This indicates that our model can exploit multi-view information in more effectively. Table 4.5 shows the comparisons on the 3DHP dataset. As can be seen, the MPJPE and PMPJPE of our method reach 102.7 and 70.1 respectively, outperforming previous methods. The PCK of our model is 78.7,

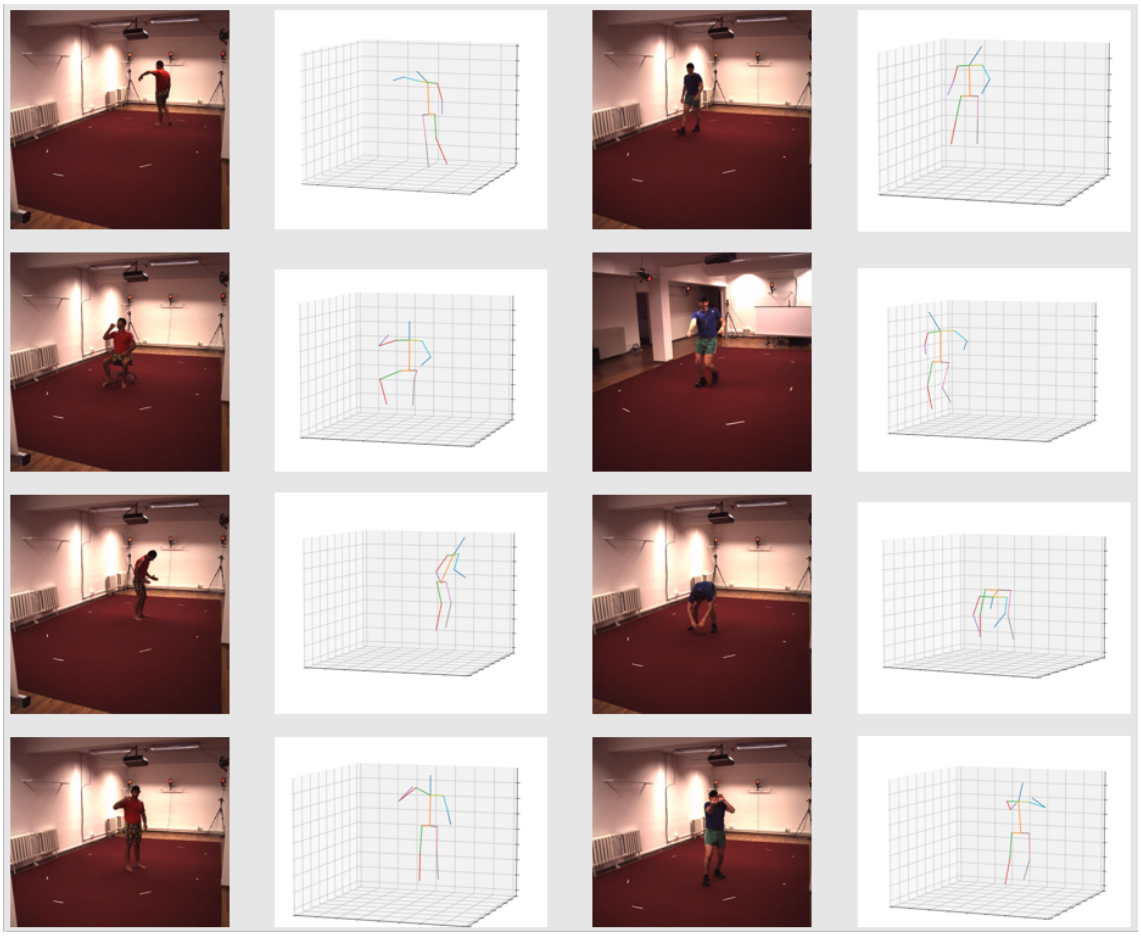

Figure 4.3 : Quantitative results of our method on the H36M dataset. To demonstrate the Generalization Ability of our method, the model is trained on the 3DHP dataset first and then test on the H36M dataset.

Table 4.5 : Comparison of evaluation on the 3DHP dataset. We present the MPJPE error, PMPJPE error and PCK for recent state-of-the-art weakly/self-supervised methods. The MPJPE error and PMPJPE error are given in mm, PCK error is given in %.

	MPJPE	PMPJPE	PCK
Rhodin et al.[95]	121.8	-	72.7
Kocabas et al.[64]	125.7	-	64.7
Chen et al.[13]	-	-	71.1
Kundu et al.[68]	103.8	-	82.1
Kolotouros et al.[65]	124.8	-	66.8
Li et al.[71]	-	-	74.1
Wandt et al.[115]	104.0	70.3	77.0
Iqbal et al.[56]	109.3	107.2	**79.5**
Ours	**102.7**	**70.1**	78.7

better than most state-of-the-art methods.

4.5 Conclusion

In this work, we propose a novel approach for 3D human pose estimation in the self-supervised training setting. We design our model as a two-stage structure. The first stage estimates the 2D pose with a 2D pose estimator while the second stage lifts the 2D into 3D output. The approach explores multi-view consistency, which can solve the ambiguity problem more effectively than previous methods. The model consists of two branches. The first branch trains a 2D-3D lifting network

while the second branch utilizes multi-view information from three camera views. Besides, our method only applies multi-view settings during training and back to the monocular setting at evaluation. We also set up a pre-train scheme to help with training. For self-supervised learning without annotations, the generated 3D poses are difficult to converge due to different camera systems. We use an additional lifting network to predict the human center in 3D space, then use it to locate the person. We have designed targeted experiments for extensive ablation studies. The results demonstrate the effectiveness and robustness of our approach. The experiments on the H36M and 3DHP datasets have achieved state-of-the-art performance compared to other self-supervised methods.

Chapter 5

3D Human Pose Estimation using mmWave Radar

5.1 Introduction

Thanks to the development of deep learning algorithms, computer vision (CV) can provide exciting findings about real-world visual representations [37, 106]. Such research mainly employs vision sensors, such as cameras (including RGB and RGBD cameras) and infrared sensors, and machine learning methods are used for various applications, including object detection, object tracking, and autonomous vehicles [66, 79, 87, 24, 93]. In recent years, the CV community has also been discussing an interesting issue, that is, the assessment of body posture. The acquisition of human posture is the key to human-computer interaction. Its focus is determining various parts of the body, such as ankles, shoulders, and wrists. Its applications are rapidly expanding, including automating patient monitoring systems owing to a current lack of nurses worldwide [84]. The tracking system can also effectively monitor autonomous or semi-autonomous vehicles and assist defense forces in making correct preventive decisions based on information on enemy actions.

At present, most skeletal pose estimations use optical sensors such as cameras or infrared (IR) devices. However, the light sensor can be affected in low light and in the presence of obstacles and harsh weather conditions such as rain, fog, or snow. Sensor failure owing to insufficient light or excessive exposure has resulted in pedestrian casualties [84]. In addition, when applied to patient monitoring systems, one of the biggest challenges is the increasing privacy concerns of users.

A radio frequency-based radar sensor uses its own signal to illuminate the target, thus ensuring that it works under different lighting and harsh climate conditions. However, in contrast to traditional vision sensors, radar uses only point clouds rather than true color images. Therefore, radar plays an important role in the application of target location. In addition, using only point cloud data, it is necessary to classify the target, which is more challenging owing to a lack of radar data that can be identified.

Traditional radar systems are bulky and expensive and are therefore mainly used in commercial and defense fields. However, the continued development of microelectronics and production technologies, including RFICs, has significantly decreased the cost of sensors, allowing them to become more practical tools. Millimeter-wave radar is one example of such technology with low power, a compact form, and easy deployment. In addition, millimeter-wave radar can provide us with a high-definition point cloud display and is therefore already an important sensor for small, unmanned robots in the commercial sector, such as unmanned vehicles. The higher working frequency band also enables the millimeter-wave radar to outline the human body without extracting the facial features, thereby protecting the personal privacy of the users.

In the last two chapters, we used two different settings: multi-view, multi-person setting and multi-view, single-person, self-supervised setting. Both of them used only camera images to detect a person's skeleton. We begin to consider adding radar signals because of the drawback of the camera images we mentioned previously. We still use a self-supervised learning model, because in that way the pressure of detecting data will be small and the training network will become very easy. We propose a novel approach that uses a self-supervised training approach to estimate 3D humans using both mmWave radar and camera images with the following three highlights:

1. We propose a two-stream self-supervised approach to extracting 3D skeletons and their key-points from radar signals. The model is trained using collaborative camera and radar data, which enhance the robustness of the network.

2. The proposed model is self-supervised, which means it does not need the ground-truth label during training. The camera and radar streams are both used at the training stage, and only radar signals are used in the evaluation.

3. We collected a small dataset for the training, and we then generated the 3D labels using multi-view algorithms for the evaluation. Our evaluations of the collected data demonstrate the effectiveness and robustness of our approach.

5.2 Related work

Camera-based Human Pose Estimation

In research conducted on computer vision, a human body posture assessment is an important aspect that can be used to determine the various important joints of the human body. By locating such important joints of the human body, the skeleton of the human body can be properly described, thereby providing more references for other computer vision tasks.

For a single image, a convolutional network is used to locate the important nodes of the human body. For example, the Deep Pose [110] method uses a coarse-to-fine approach to output the coordinates directly. The Stack Hourglass network uses a typical encoder-decoder architecture. This method applies a residual approach to store images in layers, improving the images' multi-scale resolution. Like a CPM, intermediary supervision is required at each link to prevent gradual changes. The stack hourglass algorithm achieves a higher accuracy and computing speed. Since it was first proposed, many personal pose estimation methods have used a stacked hourglass structure, including structured feature learning [17], advanced Posenet

[16], and CPF [125].

However, because data acquired by a monocular camera are used, the above-mentioned method can estimate a 2D skeletal pose. To aid in the estimation of 3D skeletal poses, the HumanEva dataset employs a ring-shaped synchronized camera and a ViconPeak motion capture system to obtain real-time conditions on the ground using reflective markers placed on objects.

5.2.1 Wireless Sensing

In recent decades, we have witnessed many uses of wireless technology to track and detect human behavior. Some device-based systems must have specific wireless devices, such as mobile phones [123]. Some existing studies have employed multi-location WiFi and mobile phones [1]. The limited accuracy makes it impossible for the system to conduct tasks such as bone capture. Other devices do this by simply analyzing the radio signals of the body. RF signals [126, 127] have recently been used to estimate a human pose. They can also detect certain special movements, such as falling. In this implementation, more antennas and a larger device are employed by customizing the RF transmission device.

Although there have been many methods for detecting human behavior using radar signals, the estimation and tracking of skeletal poses is still a new topic in the fields of radio frequency and radar. On the one hand, this is due to the inability to obtain live data from public ground-based radar. MIT's CSAIL laboratory is a pioneer in RF-based attitude estimation technology. RF-Capture is the first approach to use FMCW signals to identify multiple body parts through a side wall, using FMCW signals to "stitch" them into a rough frame [126]. This was followed by the RF-Pose and RF3D frameworks, using 1.8-GHz (5.4-7.2 GHz) FMCW signals to estimate 2D and 3D skeletal poses by applying longitudinal and lateral antenna array structures, and radar heatmaps for 2D and 3D skeletal poses, respectively.

Finally, through-wall pose imaging using a 3.3-10 GHz FMCW antenna array was proposed to estimate 15 skeletal key-points with a combination of a CNN, region proposal network, and recurrent neural networks (RNNs) [104].

In this study, we propose a self-supervised training method using both radar signals and camera information to alleviate the above challenges. Instead of the explicit 3D ground-truth, we use a self-supervised method, which only requires 2D pose estimations as inputs that can be usually generated by camera images or radar signals. Our method overcomes the problem of using a camera alone and enhances the robustness of the whole system by utilizing radar data. In our design, the training does not require a large open dataset, and any small dataset collected without ground-truth labels can be applied. In the following section, we introduce our model in greater detail.

5.3 Method

We propose a system that estimates 3D human poses based on both RF signals and camera images. Our method follows a two-stream structure: The first stream takes input from the images captured by a camera, and the second stream takes input from the radar signals. Fig. 5.1 shows the structure of our model.

For both streams, we first estimate the 2D poses from the inputs. To the camera stream, we apply an off-the-shelf 2D pose estimation network to predict the 2D pose. For each frame, we crop the images using the bounding boxes detected by an available detector, where the cropped image of the camera view is denoted by I. The cropped images are then fed into the 2D pose estimator. To the radar stream, we generate the human point cloud and obtain the 2D pose. In the next step, we lift these detections into 3D poses.

The architecture of the lifting network w^v is designed based on the approach in

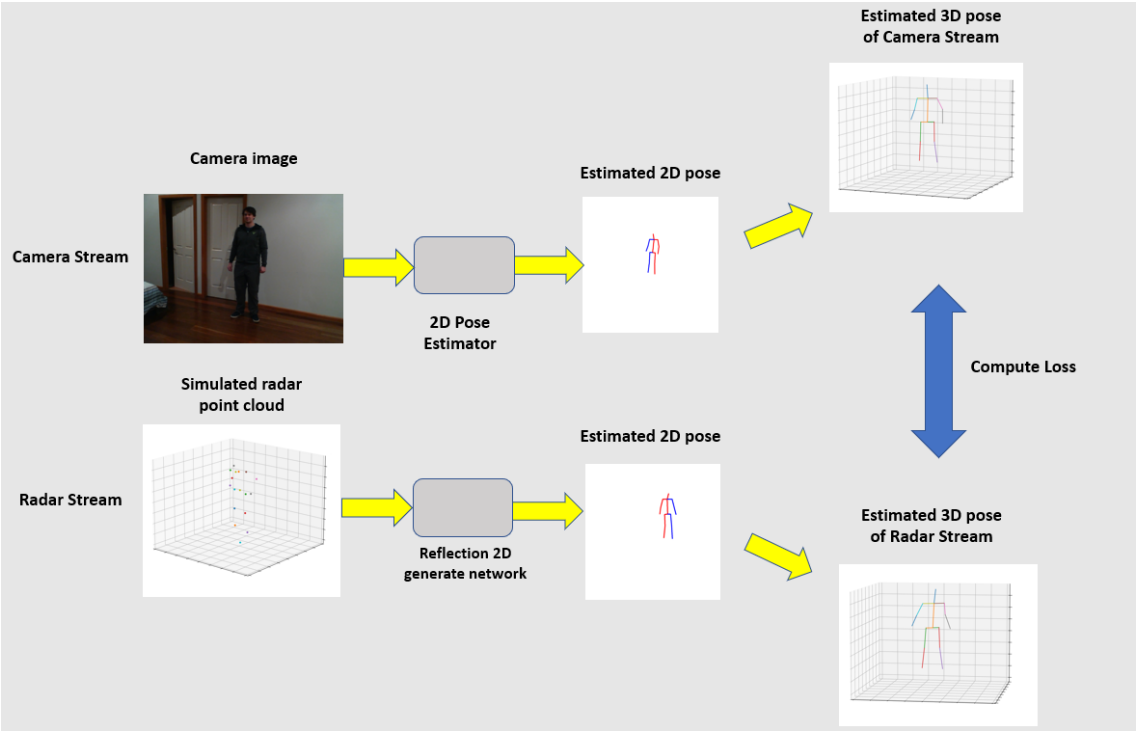

Figure 5.1 : The architecture of our proposed model. The entire network consists of two streams. The first stream takes input from the images captured by a camera, and the second stream takes input from radar signals. The camera stream applies a 2D estimator to obtain the 2D skeleton. The radar stream first generates a point cloud, reflecting it into 2D, and then uses a neural network to train a 2D skeleton. After that, the 2D skeletons are sent to a lifting network separately to obtain a 3D skeleton. To converge the network, we compute the distance between the 3D estimations from two streams.

[75]. The purpose of this lifting network is to estimate the joint positions of the human body in a 3D space with only 2D input. This network is based on batch normalization, deletion, restoration of the linear units, and residual connections. The input layer uses coordinates of N (17 in this case), and a fully connected layer is applied, which has 1024 output channels. Next, four blocks having the remaining connections are used. Each block consists of two complete layers, followed by a batch of normalized and rectified linear units and multiple outputs.

We denote $X \in \mathbb{R}^{N\times 2}$ as N detected 2D joints of the camera image, and $\tilde{X} \in \mathbb{R}^{N\times 2}$ as the estimated 2D joints of the radar. Then, the 3D lifting network predicts the 3D poses $Y \in \mathbb{R}^{N\times 3}$ and $\tilde{Y} = \{\tilde{Y}^c \in \mathbb{R}^{N\times 3} | c = 1, 2, 3\}$. Following an agreement with a previous study, we estimate a zero-point 3D pose in which Y and $\tilde{Y}$ are the 3D coordinates of a fixed root.

5.3.1 Radar Point Cloud Generation

Millimeter-wave point clouds are generally generated using frequency-modulated continuous wave (FMCW) radar with multiple transmit (Tx) and receive antennas (Rx) [98, 133]. We need precise spatial information, such as the distance and angle of the body. To allow both types of radar to work concurrently, the speed of one of the targets is necessary.

FMCW radar emits a signal called a chirp. Each chirp is a sine wave whose frequency varies linearly with time. The sweep interval of the frequencies is called the B-bandwidth. After the antenna of Tx emits a chirp, the antenna of Rx receives a chirp from the subject. The reflected wave is a delayed version of the original signal. The delay τ is proportional to the frequency difference Δf. In this way, we can estimate the distance d of the detected target to the radar:

$$d = \frac{\tau c}{2}. \tag{5.1}$$

In FMCW radar, multiple antennas are used to estimate the angle of the target. Here, Δ, the difference owing to the positions of the two RX antennas, results in a phase change ω:

$$\omega = \frac{2\pi\Delta d}{\lambda}, \tag{5.2}$$

where λ is the wavelength. We then estimate the angle of the object:

$$\theta = \sin^{-1}\left(\frac{\lambda\omega}{2\pi d}\right). \tag{5.3}$$

Using two independent chirps, the velocity of the detected target can be estimated. The measured phase difference ω is due to the moving distance vT_c of the target. Therefore, the velocity of an object can be calculated as

$$v = \frac{\lambda\omega}{4\pi T_c}. \tag{5.4}$$

On this basis, a constant false alarm rate (CFAR) is used for noise reduction, and a high-quality point cloud is obtained in the following format:

$$P_i = (x_i, y_i, z_i, v_i, I_i), i \in \mathbb{Z}^+, 1 \leq i \leq N, \tag{5.5}$$

where x_i, y_i, z_i are the 3D coordinates of the key-points, v_i represents the velocity, I_i notes the signal intensity, and N represents the number of detected key-points of each frame.

5.3.2 Training Radar 2D Pose

After generating the 3D point clouds, the next step is to estimate the 2D pose of the corresponding view. However, there are two problems with directly using 2D reflections. First, the number of joints of the 3D point cloud of each frame differs, and thus it is difficult to input them into the lifting network. Second, the point cloud is not sufficiently accurate to be used for training. To solve the above problems, we

apply a pre-training scheme to help locate the 2D poses of the point cloud. First, we reproject the point cloud into a 2D point cloud, and we then connect every two joints of the joint sets and generate a rough heatmap. We input the heatmap into a neural network to estimate the 2D pose and then lift the 2D pose into 3D.

The architecture of the lifting network w^v is designed with an inspiration by [75]. The goal of the lifting network is to estimate body joint locations in the 3D space given only a 2D input. The network is based on batch normalization, dropout, and Rectified Linear Units, as well as residual connections. The input layer takes the coordinates of N(in our case, 17) human key-points and applies a fully connected layer with 1024 output channels. It is then followed by four blocks with residual connections. Each block consists of two layers. Each layer is followed by batch normalization, rectified linear units, and dropout. Final features output by the last residual block is fed into a linear layer to get 3D poses Y:

$$Y = w^v\left(X\right). \tag{5.6}$$

5.3.3 Loss Function

Thanks to the soft-argmax algorithm, the algorithm is able to transfer the gradient flow from the 3D pose output to the input camera image such that the gradient flow is not interrupted. Based on this, we employ two loss functions to train the network: the 3D mean squared error (MSE) loss and a 2D reprojection loss. The MSE loss is common in different training tasks. One reason for using a reprojection loss is to place further constraints on the multi-view consistency. Two-dimensional detection technology can realize various possibilities of three-dimensional poses in different directions without depth information. A 2D reprojection helps in obtaining information from other views while maintaining the correct orientation.

The MSE loss is defined as follows:

$$L_{\text{mse}}^{\text{3D}} = \sum_{k=1}^{K} \frac{1}{|Y^{(k)}|} \|Y^{(k)} - \tilde{Y}^{(k)}\|_F^2. \tag{5.7}$$

The 2D re-projection loss is defined as

$$L_{\text{repj}}^{\text{2D}} = \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{1}{|y_c^{(k)}|} \|\tilde{y}_c^{(k)} - y_c^{(k)}\|_F^2. \tag{5.8}$$

5.4 Experiments

5.4.1 Dataset

We collected a variety of different data to achieve a 3D bone synchronization with radio waves. Our profiles include different human activities, including walking, sitting, shaking hands, using a mobile device, chatting, and waving. Each human motion contains around 200 images. Our model is designed to be a self-monitoring training program, and thus we do not need real 3D data for training. However, to evaluate the accuracy of the 3D skeleton, we must also have a data marker, and thus we use a multi-view camera system to generate the 3D markers.

We set up a two-camera system and build 3D labels from it. We illustrate the operation of the system through the following steps:

1. Camera calibration: We set each camera at an angle of 90 degrees. The cameras are synchronized using a standard multi-camera calibration technique relative to the global coordinate system. Once installed, a camera can capture the same person from multiple angles.

2. 2D skeleton: The next step is to use an image captured by a camera to generate a 2D skeleton. To achieve this, we use a computer vision system called OpenPose, which can display the 2D skeleton of an object based on a specific

image. Note that we use a 2D evaluation for the viewpoints of both cameras to obtain the 2D bones.

3. 3D triangulation operation: When we obtain multiple 2D skeletons from the same person, we can conduct a 3D stereoscopic measurement of their key parts through the triangulation method and obtain the corresponding 3D skeletons. We use its 2D projection X^i in space to estimate the 3D position of a particular key-point Y. This minimizes the sum of the distances from all such 2D projections:

$$\boldsymbol{Y} = \arg\min_{Y} \sum_{i \in I} \left\| C_i \boldsymbol{Y} - \boldsymbol{X}^i \right\|_2^2, \tag{5.9}$$

where C^i is a matrix that transforms the full coordinates into those of the camera view i.

5.4.2 Implementation Details

The off-the-shelf 2D pose detection backbone is the same as [15] with pre-trained weights, which outputs heatmaps and connects to a soft-argmax function to obtain the 2D poses. We use Pytorch to complete the model. We used Boost mmWave radar, which is equipped with vertical and horizontal antennas. Wireless transmission of 60-GHz FMCW chirps was applied. The transmit power was less than 1 mW. The number of batches was set to 64. We adopted the Adam optimization method and set both the weighted attenuation coefficient and the weighted attenuation deviation coefficient to 0.0005. On this basis, the learning rate was 0.0007. The total training sessions numbered 1000.

5.4.3 Ablation Study

Analysis of Different Backbone Networks

Many studies have relied on specific backbones for their performance. To demonstrate that our model does not depend on any particular baseline, in this part, we

choose different lifting backbones to evaluate our model. The residual linear [75] approach has been a commonly used method in previous studies, and Temporal Dilated [86] is the latest version that can utilize the temporal information. The comparison between them is shown in Table 5.1. From the table, we can see that our model achieves competitive results using only a simple ResLinear backbone. Therefore, the high performance of our model is not merely owing to the backbone used. In addition, the results show that our model can gain improvements when using a better backbone, which illustrates that our proposed method is suitable for any better lifting network architecture to obtain a better performance.

Table 5.1 : Comparison evaluation on different lifting backbones. The results are presented as the average error (cm) between the predictions and labels.

Backbones	Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts
RL	9.7	10.7	9.6	9.8	10.7	10.5	10.1
Ours	9.2	10.1	9.0	9.4	8.7	8.5	9.3
TD	8.6	7.8	9.3	10.5	10.4	7.3	8.7

Analysis of Different Environment Setting

We evaluated our system under different environments to test its robustness. For a traditional camera-based pose estimation, images taken under low illumination make it extremely difficult to extract the features. We evaluated our model under different illumination conditions to test the performance of the radar detection. We captured a subset consisting of low-illumination images for a second testing set. The comparison is shown in 5.2. As we can see in the results, the average error between the predictions and the labels for the high illumination test set is 9.3, and the average error for the low illumination test set is 9.6. These results show that the entire model is not dependent on the illumination conditions, which is mostly

because of the advantages of the radar-based design. This demonstrates the ability of radar to use its own signal to illuminate the target, ensuring that the system operates under low-lighting or within a harsh climate.

5.4.4 Evaluation and Discussion

The trained model was evaluated based on the test set. The testing data were used to demonstrate the performance of our model. We used the MAE between the 3D estimations and the generated 3D labels of all key-points. The results are shown in Table 5.3.

Table 5.2 : Comparison evaluation for the low-illumination and high-illumination test sets. In high illumination condition, all data are captured with the lights on, while low illumination data are captured with the lights off. The results are presented as the average error (cm) between the predictions and labels.

Illumination	Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts
High	9.2	10.1	9.0	9.4	8.7	8.5	9.3
Low	8.5	7.7	9.3	10.5	10.4	7.9	9.6

Overall, the model performs well, estimating the skeletons and markers at a small distance. There are three reasons why the network achieves such results. First, neural network models are more efficient than hand-crafted models because they can capture dependencies unknown to the designer. Second, the model can not only capture the information of radar signals but also the shape and connection of various parts of the human body. This is due to the fact that it is trained using a camera, and thus it can abstract the connection of the key-points. Third, our approach is based on time and space. In this way, the model can learn how each key-point is moving, and then use this information to determine their position, even

if they are blocked.

Table 5.3 : Average key-point localization error (cm) of the 3D skeleton prediction based on the test set.

Axis	Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts
X	8.5	9.7	10.4	7.3	8.6	9.2	8.7
Y	8.5	7.7	9.3	10.5	11.4	6.9	9.3
Z	10.4	11.7	7.9	10.4	8.3	9.6	9.8

5.5 Conclusion

This paper proposed a novel self-supervised model for learning a 3D skeleton from radar signals. Our model consists of two main components, (1) a camera stream used to capture camera images for 3D pose lifting and (2) a radar stream that first collects the point cloud and learns a 3D pose through a neural network. Our model was designed in a self-supervised manner, and thus we use the camera and radar data to help train each stream. The camera stream is only applied during the training part, whereas during the testing, only radar signals are used. Comprehensive experiments on the collected dataset demonstrate the effectiveness and robustness of our approach.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Pose estimation is a major research direction in computer vision and machine learning. It is widely used in film production and human-computer interactions, among other applications. In addition, many other computer vision problems, such as image recognition, rely on pose estimation. Pose estimation has achieved a significant improvement based on test results in recent years; however, we still have many problems to be solved for its use in practical applications. This thesis mainly considered a 3D human pose estimation algorithm based on a deep learning model. According to the specific changes in the setting, a set of corresponding models was proposed.

Three models are proposed with strong relations. The first paper solves a widely useful setting which is multi-view multi-person detection. While it performs well, it needs lots of data to train the network. Then we proposed the second model which is a self-supervised model that does not need the ground truth data. Then we found that under self-supervised conditions, the camera images have strong limitations like bad performance under low illumination conditions. To overcome those problems we add radar signals to train the network in the third model. The main contributions of our work are summarized as follows:

- We proposed a novel end-to-end training scheme for multi-view multi-person 3D pose estimation. A multi-view 2D human pose dynamic matching algorithm was designed for identifying people in different camera views. In ad-

dition, a multi-view 2D human pose dynamic matching algorithm was also proposed. This algorithm can dynamically match the corresponding 2D poses detected in multiple views for each person involved. The approach does not require knowing the exact number of people on the scene and can handle cases where false detections and severe occlusions occur.

- We presented a two-branch self-supervised approach in a multi-view training setting to train a 2D-3D neural network without the 3D ground-truth labels. The entire model only relies on the geometrical information for building the supervision signals. This chapter explored the use of 3D pose estimation with no labels.

- We proposed a two-stream self-supervised approach to extracting 3D skeletons and their key-points from radar signals. The model is trained using collaborative camera and radar data, which enhances the robustness of the network. This chapter is a further exploration of the no-label training of a 3D human pose, which makes use of radar to enhance the generalization ability of the system.

6.2 Future Work

Many research directions are being considered. As one direction, we can place both learning and reasoning algorithms in a common framework. We believe that this powerful model can directly learn geometric principles and derive 3D human poses without a calibrated system. Another direction is to leverage studies on domain adaptation/generalization to further explore how new real unlabeled 3D data can be understood using available synthetic labels. For self-supervised learning, there are some aspects we intend to study. As an example, we can attempt to transform non-labeled images into a 3D pose through cross-modal training using techniques such as synchronous and cross-genetic algorithms. Another attempt would

be the use of external datasets and temporal information to learn how to predict a temporally consistent 3D pose and therefore make the network more practical for use in the real world. In addition to the above directions, there are several areas highly related to real-world applications that we are interested in exploring:

- We would like to learn how to adapt a learned 3D human pose estimation model to a new camera system and be able to efficiently update a previously trained model with the latest data at a small cost.
- In a learned human-computer interaction with selective marking, we would like to establish virtual human pose data and thus solve the blank of the virtual human field.
- We would like to learn how to apply the trained model under a real-time daily life scenario. For example, how can a small device be used to fill out an application form?

Bibliography

[1] F. Adib, Z. Kabelac, and D. Katabi, "{Multi-Person} localization via {RF} body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, 2015, pp. 279–292.

[2] S. Amin, M. Andriluka, M. Rohrbach, and B. Schiele, "Multi-view pictorial structures for 3d human pose estimation." in *Bmvc*, vol. 2. Citeseer, 2013, p. 7.

[3] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic, "3d pictorial structures for multiple human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1669–1676.

[4] ——, "3d pictorial structures revisited: Multiple human pose estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 1929–1942, 2015.

[5] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr, "A study of parts-based object class detection using complete graphs," *International journal of computer vision*, vol. 87, no. 1-2, p. 93, 2010.

[6] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2544–2550.

[7] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.

[8] M. Burenius, J. Sullivan, and S. Carlsson, "3d pictorial structures for multiple view articulated pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3618–3625.

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.

[10] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4733–4742.

[11] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[12] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7035–7043.

[13] C.-H. Chen, A. Tyagi, A. Agrawal, D. Drover, S. Stojanov, and J. M. Rehg, "Unsupervised 3d pose estimation with geometric self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5714–5724.

[14] X. Chen, K.-Y. Lin, W. Liu, C. Qian, and L. Lin, "Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation,"

in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 895–10 904.

[15] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7103–7112.

[16] Y. Chen, C. Shen, X.-S. Wei, L. Liu, and J. Yang, "Adversarial posenet: A structure-aware convolutional network for human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1212–1221.

[17] X. Chu, W. Ouyang, H. Li, and X. Wang, "Structured feature learning for pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4715–4723.

[18] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[19] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.

[20] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6638–6646.

[21] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of*

the IEEE conference on computer vision and pattern recognition, 2014, pp. 1090–1097.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[23] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool, "Temporal 3d convnets: New architecture and transfer learning for video classification," *arXiv preprint arXiv:1711.08200*, 2017.

[24] V. N. Dobrokhodov, I. I. Kaminer, K. D. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small uavs," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[25] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*. IEEE, 2005, pp. 65–72.

[26] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[27] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou, "Fast and robust multi-person 3d pose estimation from multiple views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7792–7801.

[28] S. Ershadi-Nasab, E. Noury, S. Kasaei, and E. Sanaei, "Multiple human 3d pose estimation from multiview images," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15 573–15 601, 2018.

[29] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[30] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2334–2343.

[31] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[32] R. Faster, "Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 9199, no. 10.5555, pp. 2 969 239–2 969 250, 2015.

[33] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[34] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[35] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp.

55–79, 2005.

[36] H.-Y. Fish Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, "Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4354–4362.

[37] D. Forsyth and J. Ponce, *Computer vision: A modern approach.* Prentice hall, 2011.

[38] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2013.

[39] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[40] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[41] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt, "In the wild human pose estimation using explicit 2d features and intermediate 3d representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 905–10 914.

[42] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2015.

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

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[45] ——, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European conference on computer vision*. Springer, 2016, pp. 749–765.

[47] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*. Springer, 2012, pp. 702–715.

[48] ——, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[49] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *International conference on learning representations*, 2018.

[50] V.-T. Hoang and K.-H. Jo, "3-d human pose estimation using cascade of multiple neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2064–2072, 2018.

[51] C. Huang, S. Jiang, Y. Li, Z. Zhang, J. Traish, C. Deng, S. Ferguson, and R. Y. D. Xu, "End-to-end dynamic matching network for multi-view

multi-person 3d pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 477–493.

[52] S. Huang, M. Gong, and D. Tao, "A coarse-fine network for keypoint localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3028–3037.

[53] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "Deepercut: A deeper, stronger, and faster multi-person pose estimation model," in *European Conference on Computer Vision*. Springer, 2016, pp. 34–50.

[54] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[55] U. Iqbal and J. Gall, "Multi-person pose estimation with local joint-to-person associations," in *European Conference on Computer Vision*. Springer, 2016, pp. 627–642.

[56] U. Iqbal, P. Molchanov, and J. Kautz, "Weakly-supervised 3d human pose learning via multi-view images in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5243–5252.

[57] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," *arXiv preprint arXiv:1905.05754*, 2019.

[58] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh, "Panoptic studio: A massively multiview

system for social motion capture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3334–3342.

[59] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews *et al.*, "Panoptic studio: A massively multiview system for social interaction capture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 190–204, 2017.

[60] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[61] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

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

[63] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[64] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," *arXiv preprint arXiv:1903.02330*, 2019.

[65] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2252–2261.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[67] Y. Kudo, K. Ogaki, Y. Matsui, and Y. Odagiri, "Unsupervised adversarial learning of 3d human pose from 2d joint locations," *arXiv preprint arXiv:1803.08244*, 2018.

[68] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty, "Self-supervised 3d human pose estimation via part guided novel image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6152–6162.

[69] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2, pp. 107–123, 2005.

[70] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 332–347.

[71] Y. Li, K. Li, S. Jiang, Z. Zhang, C. Huang, and R. Y. Da Xu, "Geometry-driven self-supervised method for 3d human pose estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 442–11 449.

[72] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[73] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[74] B. D. Lucas, T. Kanade *et al.*, *An iterative image registration technique with an application to stereo vision*. Vancouver, 1981, vol. 81.

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

[76] 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 *2017 international conference on 3D vision (3DV)*. IEEE, 2017, pp. 506–516.

[77] 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 *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 120–130.

[78] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *Acm transactions on graphics (tog)*, vol. 36, no. 4, pp. 1–14, 2017.

[79] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern analysis and applications*, vol. 8, no. 1, pp. 17–31, 2005.

[80] G. Moon, J. Y. Chang, 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 *Proceedings of the IEEE conference on computer vision and pattern Recognition*, 2018, pp. 5079–5088.

[81] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2823–2832.

[82] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[83] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi, "C3dpo: Canonical 3d pose networks for non-rigid structure from motion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7688–7697.

[84] J. A. Oulton, "The global nursing shortage: an overview of issues and actions," *Policy, Politics, & Nursing Practice*, vol. 7, no. 3_suppl, pp. 34S–39S, 2006.

[85] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7025–7034.

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

[87] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2, pp. 123–139, 2009.

[88] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, "Deepcut: Joint subset partition and labeling for multi person pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4929–4937.

[89] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7699–7707.

[90] M. Rayat Imtiaz Hossain and J. J. Little, "Exploiting temporal information for 3d pose estimation," *arXiv e-prints*, pp. arXiv–1711, 2017.

[91] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[92] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[93] R. Reulke, S. Bauer, T. Doring, and F. Meysel, "Traffic surveillance using multi-camera detection and multi-target tracking," in *Image and Vision Computing New Zealand*, 2007, pp. 175–180.

[94] H. Rhodin, V. Constantin, I. Katircioglu, M. Salzmann, and P. Fua, "Neural scene decomposition for multi-person motion capture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7703–7713.

[95] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua, "Learning monocular 3d human pose estimation

from multi-view images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8437–8446.

[96] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net++: Multi-person 2d and 3d pose detection in natural images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1146–1161, 2019.

[97] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[98] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 032–10 044, 2020.

[99] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014.

[100] ——, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[101] Y. Song, L.-P. Morency, and R. Davis, "Multimodal human behavior analysis: learning correlation and interaction across modalities," in *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 2012, pp. 27–30.

[102] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4597–4605.

[103] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2602–2611.

[104] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 529–545.

[105] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[106] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

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

[108] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2500–2509.

[109] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *Advances in neural information processing systems*, vol. 27, 2014.

[110] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[111] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of*

the IEEE international conference on computer vision, 2015, pp. 4489–4497.

[112] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognition Letters*, vol. 49, pp. 250–258, 2014.

[113] S. Vosoughi and M. A. Amer, "Deep 3d human pose estimation under partial body presence," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 569–573.

[114] B. Wandt and B. Rosenhahn, "Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7782–7791.

[115] B. Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn, "Canonpose: Self-supervised monocular 3d human pose estimation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 294–13 304.

[116] C. Wang, C. Kong, and S. Lucey, "Distill knowledge from nrsfm for weakly supervised 3d pose learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 743–752.

[117] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[118] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3d human pose machines with self-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1069–1082, 2019.

[119] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE*

transactions on pattern analysis and machine intelligence, vol. 41, no. 11, pp. 2740–2755, 2018.

[120] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[121] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 748–756.

[122] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[123] J. Xiong and K. Jamieson, "{ArrayTrack}: A {Fine-Grained} indoor location system," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[124] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5255–5264.

[125] H. Zhang, H. Ouyang, S. Liu, X. Qi, X. Shen, R. Yang, and J. Jia, "Human pose estimation with spatial contextual information," *arXiv preprint arXiv:1901.01760*, 2019.

[126] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7356–7365.

[127] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, "Rf-based 3d skeletons," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 267–281.

[128] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1318–1327.

[129] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5157–5166.

[130] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4966–4975.

[131] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3d human pose estimation in the wild: a weakly-supervised approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 398–407.

[132] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *European Conference on Computer Vision*. Springer, 2016, pp. 186–201.

[133] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.