UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Deep Learning based Human Pose Estimation

by

Yang Li

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

Doctor of Philosophy

Sydney, Australia

2020

Certificate of Authorship/Originality

I, Yang Li declare that this thesis, is submitted in fulfilment of the requirements for the award of Ph.D, in the Faculty of Engineer and Information Technology 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.

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with Beijing Institute of Technology.

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

Signature: Production Note:
Signature removed prior to publication.

Date: 30/09/2020

ABSTRACT

Deep Learning based Human Pose Estimation

by

Yang Li

Human pose estimation is an important research area in vision-based human activity analysis. Human pose estimation aims to estimate the human articulate joint positions in 2D/3D space from images or videos. Due to the complexity of the real environment and the diversity of human poses, vision-based human pose estimation is challenging. Recently, the rapid development of deep learning has much promoted the simulation of the analysis and reasoning capabilities of the human visual system. Therefore, it is of considerable significance to further explore vision-based human pose estimation using deep learning techniques. Specifically, this thesis proposes a series of methods for human pose estimation, summarized as follows:

We propose a video-based 2D pose estimation model, which embeds a multi-scale TCE module into the encoder-decoder network architecture for explicitly exploring temporal consistency in videos. The TCE module applies the learnable offset field to capture the geometric transformation between adjacent frames at the feature level. In addition, we explore the multi-scale geometric transformations at the feature level by integrating the spatial pyramid within the TCE module, which achieves further performance improvements.

We propose a self-supervised approach for 3D human pose estimation, which only relies on geometric prior knowledge and does not require any 3D human pose annotations. To this end, we design the transform re-projection loss, which is an effective technique to exploit multi-view consistency information and constrain the estimated 3D poses during training. Besides, we introduce a root position regression

branch to restore the global 3D poses during training. In this way, the network can reserve the scale information of re-projected 2D poses, which can improve the accuracy of the predicted 3D poses.

We propose a self-supervised 3D human pose estimation method based on the consistent factorization network, which fully disentangles the 3D human shape and camera viewpoint to overcome the projection ambiguity problem. To this end, we design a simple and effective loss function using multi-view information to constrain the canonical 3D human pose. Moreover, in order to reconstruct robust canonical 3D human poses, we represent 3D human pose as a combination of a dictionary of 3D pose basis, and adopt geometric information of 3D human poses to learn a hierarchical dictionary from 2D human poses by solving the NRSfM problem.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor A/Prof. Richard Yi Da Xu and Kan Li (Beijing Institute of Technology) for their continuous support of my Ph.D study and research, for his patience, motivation, and immense knowledge. Their guidance helped me in all the time of my research. I could not have imagined having a better advisor and mentor for my Ph.D study.

My sincere thanks also go to my fellow labmates: Shuai Jiang, Ziyue Zhang, Congzhentao Huang, Chen Deng, Ximeng Zhao, Jason, etc., for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last one year. I have the honour of studying and working with them in the past year which is valuably stamped in my life.

Finally, I must express my very profound gratitude to my parents and to girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and the process of researching. This accomplishment would not have been possible without them. Thank you.

Yang Li

List of Publications

Journal Papers

J-1. **Li Y.**, Li K., Wang X., Xu, R. Y. D., Exploring temporal consistency for human pose estimation in videos, Pattern Recognition, 2020, 103: 107258.

J-2. **Li Y.**, Li K., Wang X., Recognizing actions in images by fusing multiple body structure cues, Pattern Recognition, 2020: 107341.

Conference Papers

C-1. **Li Y.**, Li K., Jiang S., Zhang Z., Huang C., Xu, R. Y. D., Geometry-Driven Self-Supervised Method for 3D Human Pose Estimation, AAAI, 2020: 11442-11449.

C-2. **Li Y.**, Li K., Wang X., Deeply-Supervised CNN Model for Action Recognition with Trainable Feature Aggregation, IJCAI, 2018: 807-813.

C-3. 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-4. 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.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aims and Motivations

Computer vision is a classical research topic of artificial intelligence. It aims to equip the computer with a capability to cognize and understand visual signals in the surrounding real world like human beings. With the popularity of web cameras and surveillance cameras, a large number of image and video data are being produced to record people's activity. Therefore, understanding human activity has become one of the most active research areas in the computer vision community. Human pose estimation is a fundamental task for human activity analysis. It aims to locate the human anatomical keypoints, which can further locate and reconstruct the human body and serve for the more advanced analysis of visual signals.

Besides, human pose estimation is also widely used in fields like video surveillance, human-computer interaction, sports analysis, virtual reality, animation generation, etc. For example, human pose estimation can be used to track human subjects' motion for interactive gaming. Popularly, Kinect, launched by Microsoft, used 3D pose estimation to track the motion of the human player and to use it to render the action of the virtual character. As for sports analysis, human pose estimation can reconstruct the athlete's motion from the daily training videos. The analyst can then conduct quantitative analysis combined with human physiology, physics, and other domain knowledge, which is more scientific and gets rid of purely relying on experience. As for the massive surveillance video data, human pose estimation can automatically detect and localize human bodies in surveillance videos, which significantly improves the speed of data processing tasks, such as abnormal behavior detection. Human pose estimation can also be used for CGI applications. Graphics, styles, fancy enhancements, equipment, and artwork can be superimposed if their human pose can be estimated. By tracking the variations of this human pose, the rendered graphics can naturally fit the person as he/she moves.

Human pose estimation can be divided into two categories: 2D human pose estimation and 3D human pose estimation. Due to the complexity of the real environment and the diversity of human poses, both 2D and 3D human pose estimation are very challenging tasks. Recently, with the rapid development of Convolutional Neural Networks (CNNs), deep learning has been increasingly exploited on the task of human pose estimation. Although deep learning based methods have achieved significant improvements, they still face some challenges. As for 2D human pose estimation, most existing methods focus on designing novel network architectures for image-based 2D pose estimation. Although these methods can be directly applied to video data, they usually obtain suboptimal performances because the direct application of image-based methods cannot leverage the rich temporal information inherent in video data. As for 3D human pose estimation, a typical neural network model needs a large amount of training data. However, annotating 3D human joint positions is an expensive process. Moreover, there are well-founded geometrical theories on how to project 2D images to 3D skeletons. Merely using a neural network to approximate this projection may lead to the network subject to overfitting training data. Toward the above issues, this thesis researches video-based 2D pose estimation and self-supervised 3D pose estimation methods. Specifically, we propose a series of methods for human pose estimation, which are summarized as follows:

1. We propose a video-based 2D pose estimation model, which embeds a multi-scale TCE module into the encoder-decoder network architecture for explic-

itly exploring temporal consistency in videos. The TCE module applies the learnable offset field to capture the geometric transformation between adjacent frames at the feature level. Compared with the existing model-based methods, it can explicitly model the temporal consistency information in an end-to-end network. Compared with the existing post-enhancement methods, it does not require additional optical flow calculations and is more computationally efficient. In addition, we explore the multi-scale geometric transformations at the feature level by integrating the spatial pyramid within the TCE module, which achieves further performance improvements. Experimental results demonstrate that the proposed network has achieved state-of-the-art performance in both accuracy and computational efficiency.

2. We propose a self-supervised approach for 3D human pose estimation, which only relies on geometric prior knowledge and does not require any 3D human pose annotations. To this end, we design the transform re-projection loss, which is an effective technique to exploit multi-view consistency information and constrain the estimated 3D poses during training. Moreover, we integrate the re-projection losses with the 2D joint confidences of different camera views to alleviate the self-occlusion problem. Besides, we introduce a root position regression branch to restore the global 3D poses during training. In this way, the network can reserve the scale information of re-projected 2D poses, which can improve the accuracy of the predicted 3D poses. Finally, a pre-training technique is designed to help the two-branch network converge. Experimental results show that this method achieves better performance compared with recent weakly/self-supervised methods.

3. We propose a self-supervised 3D human pose estimation method based on the consistent factorization network, which fully disentangles the 3D human

shape and camera viewpoint to overcome the projection ambiguity problem. To this end, we design a simple and effective loss function using multi-view information to constrain the canonical 3D human pose. Moreover, in order to reconstruct robust canonical 3D human poses, we represent 3D human pose as a combination of a dictionary of 3D pose basis, and adopt geometric information of 3D human poses to learn a hierarchical dictionary from 2D human poses by solving the NRSfM problem. The hierarchical dictionary can be learned without the need for 3D human pose annotations, and has a stronger expression ability compared with the single-level dictionary. Experimental results show that the proposed method can maximally disentangle 3D human shapes and camera viewpoints, as well as reconstruct accurate 3D human poses.

1.2 Literature Review

1.2.1 Overview

Human pose estimation, also known as human keypoint estimation, aims to locate the anatomical keypoints in the human body. It is a fundamental task in the computer vision community, which has important theoretical value and broad applications in many fields such as human activity recognition, sports analysis, and human-computer interaction. Recently, many world-class research teams and institutes have invested many resources to study this problem. For example, the Robotics Institute at Carnegie Mellon University has constructed a large-scale multi-view human motion capture system named Panoptic Studio [45]. Based on this system, researchers have proposed multiple datasets for different tasks such as human pose estimation, hand pose estimation, and social interaction analysis. Microsoft Research has constructed a large-scale 2D human pose dataset named Common Objects in Context (COCO) [53], which gathers images of complex everyday scenes containing common objects in their natural context. At the same time, they organized competitions and workshops, which greatly promoted the development of 2D pose estimation technologies. Max Planck Institute for Computer Science also proposed MPII [1] and MPI-INF-3DHP [56] datasets, which are widely used 2D and 3D human pose datasets.

Human pose estimation already becomes one of the hottest topics in top computer vision conferences and journals, such as CVPR (IEEE Conference on Computer Vision and Pattern Recognition), ICCV (International Conference on Computer Vision), ECCV (European Conference on Computer Vision), PAMI (IEEE Transaction on Pattern Analysis and Machine Intelligence), TIP (IEEE Transaction on Image Analysis), IJCV (International Journal of Computer Vision), CVIU (Computer Vision and Image Understanding) and PR (Pattern Recognition), etc. Human pose estimation can be divided into 2D pose estimation and 3D pose estimation, which estimate human joint positions in two-dimensional and three-dimensional space, respectively. In the following, we will review the relevant literature in detail.

1.2.2 2D Human Pose Estimation

In this part, we first review the representative works of 2D human pose estimation.

Image-based Approaches

Traditional methods of image-based 2D pose estimation are mostly bottom-up part-based. These methods represent the human pose as a collection of human body parts, and they adopt the deformable model to describe spatial relationships of body parts. In 1973, Fishler et al. [25] proposed the pictorial structure model for visual object representations. Subsequently, Felzenszwalb et al. [24] proposed a general object recognition model based on the pictorial structure model. Since then, the pictorial structure model was widely used in human pose estimation, and plenty of part-based methods [2, 101, 68, 69, 83, 50, 76, 94] were gradually proposed. For example, Yang and Ramanan [101] designed a general, flexible mixture model for capturing contextual co-occurrence relations between parts, augmenting standard spring models that encode spatial relations. Pishchulin et al. [68] used the poselet prior to improve the pictorial structure model. These methods mainly rely on hand-crafted features, such as Histogram of Oriented Gradient (HoG) and Scale-Invariant Feature Transform (SIFT), to detect human body parts, and then use the dynamic programming algorithm to obtain the optimal human pose configuration. However, these methods lack generalization ability for images of complex everyday scenes where human joints are truncated or severely occluded.

With the great success of deep learning in object classification and detection, researchers tried to apply Convolutional Neural Networks (CNNs) to human pose estimation. Meanwhile, the availability of large-scale human pose datasets, such as FLIC [73], MPII [1], and Microsoft COCO [53], make it possible to train deep networks. Since CNNs can learn high-level visual features through stacked convolution and pooling layers, these methods can directly predict human joint positions from input images. In 2014, Jain et al. [42] and Toshev et al. [87] proposed to train CNNs to regress human joint positions. Compared with traditional methods following the multi-stage pipeline that includes extracting hand-crafted features, training body part detectors, and modeling spatial relations of body parts, deep learning based methods can be trained in an end-to-end manner, at the same time achieve better performance for images in complex natural scenes. Mainstream deep learning based methods are dedicated to designing network architectures that are more suitable for the task of human pose estimation. For example, Wei et al. [95] proposed the Convolutional Pose Machine (CPM), which produces increasingly refined pose estimations by directly operating on belief maps from previous stages. Newell et al. [58] introduced the Stacked Hourglass Network (SHN) that improves the performance by repeating bottom-up, top-down processing. Cao et al. [10] introduced the Part Affinity Fields (PAFs) to learn the association of body parts based on the CPM architecture, which significantly outperformed previous works. Belagiannis et al. [7] designed a recurrent neural network to improve the results iteratively. All of these networks follow the multi-stage cascading architecture, which can capture large spatial context information and improve the precision of predicted joint positions through multi-stage refinement. This multi-stage architecture has achieved state-of-the-art results in many image-based benchmarks.

Besides, some works [17, 98, 32, 79, 35] explore the fusion of multi-scale features from different levels of CNNs. For example, He et al. [32] and Chen et al. [17] applied the feature pyramid structure for pose estimation by adopting the Feature Pyramid Network (FPN) [52]. Yang et al. [98] designed the Pyramid Residual Module (RPM) that learns feature pyramids using different subsampling ratios in a multi-branch network. There are also some methods [99, 86, 19, 23, 92] that combined CNNs with graphical models to learn both convolutional features and joint spatial constraints in an end-to-end network. For example, Tompson et al. [86] combined a CNN and a Markov Random Field (MRF) to exploit the spatial relationships between human joints in a unified model. Chu et al. [19] proposed a deep structured feature learning framework that models the correlations among the convolutional feature maps of body joints for accurate pose estimation. The success of all the above works demonstrates how a large spatial context is essential for CNN-based pose estimation methods.

Video-based Approaches

Although image-based methods can be directly applied to video data, they usually obtain sub-optimal performance because the direct application of image-based methods cannot leverage the rich temporal information inherent in video data. We will summarize the existing video-based methods and analyze how they explore temporal information in videos. Conventional methods [61, 59, 40, 18, 112, 26, 105] consider video temporal information by adding connections in the temporal dimension on the pictorial structure model. For example, Cherian et al. [18] cast the video-based pose estimation problem as an optimization problem defined on body parts with spatio-temporal links between frames.

Recent works attempt to integrate temporal cues in advanced deep models to improve video-based pose estimation performance. Among them, the most common methods [67, 11, 78, 97] investigate temporal context by using optical flow. As optical flow defines the distribution of apparent velocities of movement, it can help capture the geometric transformations between frames to refine the predicted heatmaps. For example, Song et al. [78] used optical flow to exploit image evidence from adjacent frames. Pfister et al. [67] utilized optical flow to align output heatmaps from neighboring frames to improve the performance of video pose estimation. Some other methods [28, 54] capture the temporal dependency using Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which have become dominating tools for sequence tasks thanks to their power in long-range temporal representation. For example, Luo et al. [54] proposed a recurrent model with LSTM to consider the temporal information for pose estimation in videos. Gkioxari et al. [28] introduced a chained model using CNNs, where the pose prediction depends not only on the input but also on the output of the previous frame. There are also methods of applying 3D convolution to learn representations of video clips. Girdhar et al. [27] inflated the 2D convolution in the Mask R-CNN into 3D, which leverages temporal information over video clips to generate more robust pose predictions in videos.

1.2.3 3D Human Pose Estimation

3D human pose estimation refers to the process of reconstructing 3D human skeletons from single-view and multi-view images. According to the inputs, 3D human pose estimation can be divided into multi-view 3D pose reconstruction and monocular 3D pose estimation.

Multi-view 3D Pose Reconstruction

Multi-view 3D pose reconstruction requires the camera intrinsic and extrinsic parameters obtained through camera calibration, and then use the 2D skeletons from multiple corresponding camera views to reconstruct 3D human skeletons. Early representative works include the 3D Pictorial Structure (3DPS) models [5, 6] and the multi-view geometry based methods [31]. 3DPS models construct a probabilistic graphical model in which the nodes represent 3D positions of human joints, and the edges encode spatial relationships between human joints. Specifically, a node's state space is usually represented by a discrete 3D grid, and the conditional probability of each position is represented by the confidence of the corresponding re-projected position on the 2D keypoint heatmap. The prior is described by the bone length constraint or the confidence detected by body part detectors. As a result, 3DPS models infer 3D human skeletons by maximizing the posterior probability. The multi-view geometry based methods apply the triangulation algorithm, which solves overdetermined equations to calculate the homogeneous coordinates of human joints. Since the 2D coordinates of human joints usually cannot be accurately estimated, Random Sample Consensus (RANSAC) and Huber loss were used to search for optimal solutions.

Some recent methods [41, 21, 85, 15, 70] introduced deep learning to improve the robustness of multi-view 3D pose reconstruction methods. For example, Iskakov et al. [41] proposed an end-to-end network that can directly reconstruct 3D human skeletons from multi-view image inputs. Dong et al[21] proposed a real-time multi-person 3D pose reconstruction algorithm that focuses on solving the person matching problem in multi-view multi-person scenarios. Besides, some researchers [72, 45] attempted to obtain accurate 3D pose annotations in a marker-less motion capture system, and further provide training data for monocular 3D pose estimation models. Although the above methods use deep learning techniques, such as 2D human pose estimation and person re-identification, the core algorithm about 3D skeleton reconstruction is still based on 3DPS or multi-view geometry.

Monocular 3D Pose Estimation

Recent researches focus on estimating 3D human poses using only single-view inputs. Thanks to the powerful fitting ability of deep neural networks and the availability of multiple large-scale 3D pose datasets such as Human3.6M [38] and MPI-INF-3DHP [56], plenty of monocular 3D human pose estimation methods have been proposed and made significant progress. Unlike multi-view methods, monocular methods can only estimate the relative 3D positions of human joints [55], and cannot obtain the absolute positions in the world or camera coordinate. 3D pose annotations are generally collected through the marker-based Motion Capture system, which is labor-intensive and expensive. Therefore, weakly/self-supervised learning paradigms have been increasingly explored in recent works, where 3D pose estimation networks can be effectively trained with part of or without 3D pose annotations.

Next, we will introduce representative works from two aspects, fully-supervised approaches and weakly/self-supervised approaches.

Fully-supervised Approaches

Fully-supervised methods concentrate on designing effective 3D pose estimation network architectures. Existing methods can be generally classified into two categories: **two-stage methods** and **single-stage methods**. Two-stage methods

[39, 84, 55, 23, 12, 51] first obtain 2D joint locations through the advanced 2D keypoint detector such as Stacked Hourglass Network (SHN) [58] and Cascaded Pyramid Network (CPN) [17], and then lifts them into 3D space through a lifting network. The second stage is the core of two-stage methods, which learns the mapping between the 2D and 3D positions of the human joints. To this end, various 2D-to-3D lifting network backbones were designed. For example, Martinez et al. [55] proposed a simple baseline using a simple neural network with only two fully connected layers and achieved surprising results. Although this network has a simple architecture, it achieved good performance and is widely used in subsequent works. Since human skeletons are with the graph-like structure, hao et al. [107] and Ci et al. [20] attempted to exploit the novel Graph Convolution Networks (GCNs) to capture the semantic relationships between human joints for accurate 3D human pose regression. Besides, there are also works [65, 3, 110, 109] that considers temporal information from frame sequence to produce more robust predictions.

The single-stage method [82, 63, 30, 81, 80, 104]directly predict depth values of human joints from monocular images through CNNs. Most of the single-stage methods [80, 56, 57] adopt the joint position regression strategy. In addition, Sun et al. [81] proposed an integral regression method, which used the *soft-argmax* operation to obtain the joint coordinate vector from the predicted heatmap in a differentiable way. Pavlakos et al. [63] discretized the 3D space around the target position, proposed a more natural 3D pose representation, and trained a CNN to predict the probability value of the voxel corresponding to each human joint.

Weakly/Self-supervised Approaches

Monocular 3D pose estimation is an inverse graphic process. In order to train the 3D pose estimation network without explicit 3D pose annotations, some works [89] introduced the prior of camera projection geometry to train the network in the weakly or self-supervised manner. Among them, the re-projection loss is the most widely used technique [46, 96, 90, 65, 93, 8], which projects the 3D skeleton predicted by the network back to the 2D space through perspective or orthogonal projection, and calculates the error between the re-projection and input 2D pose as the training loss. Here, the re-projection process can be regarded as a decoder without any trainable parameters. However, due to the projection ambiguity problem [31, 75], using re-projection loss alone cannot effectively constrain the network outputs. In order to alleviate this problem, existing methods generally adopt the following strategies:

1. **Bone Length Constraint:** In order to avoid unreasonable predictions, some works [30, 65, 72] make the predicted 3D poses satisfy kinematics by enforcing the bone length similarity between predicted and ground-truth skeletons. For example, Pavllo et al. [65] added an extra soft constraint of bone length to the objective function when optimizing the network.

2. **Adversarial Loss:** Inspired by Generative Adversarial Network (GAN) [29], adversarial loss [89, 100] is widely used to solve the projection ambiguity problem. It encourages the output 3D poses on the real human manifold by introducing a real/fake 3D skeleton discriminator. For example, Tung et al. [89] proposed the Adversarial Inverse Graphic Network (AIGN), which uses the adversarial prior to match the distribution between the predictions and a collection ground-truth for the task such as 2D-to-3D lifting and image-to-image translation. Similarly, Wandt et al. [90] proposed a weakly-supervised method with the adversarial supervision for 3D human pose estimation. Chen et al. [13] exploits the geometric self-consistency of the lift-reproject-lift process with the adversarial prior of 2D poses. These methods require some extra unpaired 3D pose annotations (without 2D-3D correspondence) to make the network

memorize the distribution of real 3D skeletons during training.

3. **Multi-view Constraint:** Also, some works [88, 71, 48, 13] introduced multi-view constraint training human 3D pose estimation network. Unlike multi-view 3D pose reconstruction, these methods only require multi-view images as input during the training phase. For example, Rhodin et al. [71] pre-trained an encoder-decoder network that predicts an image from one view to another. In this way, the network learned geometry-aware representations and then was fine-tuned using a small amount of supervision to predict 3D human poses. Kocabas et al. [48] applied Epipolar geometry to obtain "real" 3D poses from estimated 2D joint positions of multiple views and used them for training a 3D pose estimator.

4. **Dictionary-based Technique:** All of the above methods directly regress depth values or 3D positions of human joints at the corresponding camera view. Some methods advocate using a dictionary of 3D pose basis elements to represent a 3D human skeleton. These methods reduce the space of all allowed 3D coordinates, making the predictions within the "human" sub-space. For example, Tung et al. [89] adopt PCA on the orientation-aligned training set to obtain a shape dictionary, and represent a 3D human pose as a linear combination of 3D shape basis. Novotny et al. [60] also represent 3D human poses as a combination of shape basis. They consider the dictionary as the weights of a linear layer and learn it with the 3D pose estimation network in an end-to-end manner. Most recently, some researches make advances in Non-rigid Structure from Motion (NRSfM), a classical technique to reconstruct 3D shapes of articulated 3D points. For example, Kong and Lucey [49] proposed the Deep-NRSfM network architecture as solving a multi-layer block sparse dictionary learning problem, and achieve high-quality 3D human pose reconstructions.

Wang et al. [91] proposed a new knowledge distilling algorithm applicable to Deep-NRSfM based on dictionary learning, and achieves weakly-supervised learning using solely 2D human joint annotations.

1.3 Thesis Organization

This thesis mainly conducts research on deep learning based human pose estimation, and the thesis is organized as follows:

- *Chapter 2*: This chapter presents the proposed video-based 2D human pose estimation framework, which explores temporal consistency in videos to improve the performance.
- *Chapter 3*: This chapter presents the proposed self-supervised 3D pose estimation method. The method follows the two-stage pipeline, where the first stage is 2D pose estimation and adaptive to the video-based 2D pose estimation network described in Chapter 2, and the second stage lifts 2D poses into the 3D space through a 2D-to-3D lifting network.
- *Chapter 4*: This chapter presents the consistent factorization network, where 3D human shape and camera viewpoint are consistently decomposed to overcome the projection ambiguity problem. Unlike the method proposed in Chapter 3, this method does not require intrinsic and extrinsic camera parameters during training, which can be considered an extension of the method in Chapter 3.
- *Chapter 5*: A brief summary of the thesis contents and its contributions are given in the final chapter. Recommendation for future works is given as well.

Chapter 2

Exploring Temporal Consistency for 2D Human Pose Estimation in Videos

2.1 Introduction

Human pose estimation is a fundamental task in the computer vision community and has been broadly applied to many fields such as human activity recognition, sports analysis and human-computer interaction. The purpose of pose estimation is to locate the anatomical keypoints in human bodies. Previous methods have traditionally relied on hand-crafted features. They face challenges when handling unconstrained cases due to the highly articulated human body limbs, occlusion and change of viewpoint. Recently, as a result of the availability of large-scale human pose datasets [1, 53] and the rapid development of Convolutional Neural Networks (CNNs) [34], plenty of deep learning based methods for pose estimation have been proposed and achieved significant progress. Traditionally, most of these methods predict human poses from single images. Although these methods can be directly applied to video data, they usually obtain suboptimal performances because the direct application of image-based methods cannot leverage the rich temporal information inherent in video data. In this paper, we focus on improving human pose estimation in videos by fully exploring the temporal information.

Some works already attempt to integrate temporal information into the deep models to estimate human poses in videos. These works can be generally classified into two categories: The **first category** focuses on model-based methods, which adopt 3D convolution [27] or RNN [54, 28] techniques to learn spatio-temporal representations of video clips. These methods can model spatial and temporal information jointly in an end-to-end framework. However, 3D convolution and RNN have limited ability to explore the temporal consistency (e.g., geometric transformations of human body parts) between adjacent video frames. The **second category** concentrates on posterior enhancement methods [67, 11, 78, 97] that adopt optical flow to warp predicted heatmaps of neighboring frames onto that of the target frame. Since optical flow defines the distribution of apparent velocities of the movement of brightness patterns in an image [36], these methods can explicitly exploit the temporal consistency. Despite their promising results, optical flow estimation is computationally intensive and susceptible to the occlusion and motion blur problems in unconstrained videos, which affects the performance of pose estimation to some extent.

To overcome the problems arising from these methods in the above two categories, we propose a video-based pose estimation model that effectively explores the temporal consistency of videos. The core of the model is the novel Temporal Consistency Exploration (TCE) module which has major advantages over the previous model-based and posterior enhancement methods. On the one hand, the TCE can explicitly explore the temporal consistency through a learnable module. On the other hand, it is more efficient as it does not need the post-processing and extra calculation of optical flow. The TCE module captures the temporal consistency at the feature level based on the fact that the spatial information of body joint locations is well preserved in feature maps[19]. In a nutshell, the TCE module follows a recurrent architecture and predicts the geometric transformations between neighboring feature maps through the learnable offset field. Then it deforms the neighboring feature maps, and the resultant deformed feature map is combined with the original map to produce enhanced feature maps through a temporal aggregation. Moreover, since the temporal information from both forward and backward directions are complementary for predicting human joint positions, the TCE module is designed to capture temporal consistency from both directions. In addition to temporal consistency, at the same time, recent researchers found that rich spatial context has proven to play an essential role in human pose estimation [17, 10, 58]. Therefore, in our work, we further design the multi-scale TCE which tightly integrates the spatial pyramid within the TCE module. The spatial pyramid increases the receptive field of the TCE module as well as facilitating the TCE module to explore the geometric transformations at multi-scale spatial levels. Using the powerful multi-scale TCE module, we extend the encoder-decoder network architecture for exploring temporal information and achieve significant improvements in video-based pose estimation.

We comprehensively evaluate the proposed model on the public challenging datasets: sub-JHMDB [43] and Penn [106]. The results demonstrated that our model outperforms recent methods and achieves state-of-the-art performances on the two video-based pose datasets. The contributions of our work are summarized as follows:

1. In this work, we propose a video-based pose estimation model that explicitly explores the temporal consistency in videos. To achieve that, we design a novel TCE module that captures geometric transformations between frames at the feature level using the learnable offset field .

2. We explore the multi-scale geometric transformations at the feature level by tightly integrating the spatial pyramid within the TCE module, which achieves further performance improvements.

3. Our model achieves 96.4% and 99.2% average accuracy on Sub-JHMDB and Penn datasets respectively using the PCK@0.2 metric, which outperforms all recent approaches.

2.2 Related Work

2.2.1 Image-based Pose Estimation

Traditional methods for pose estimation in images mostly rely on hand-crafted features (e.g., SIFT, HOG) and seek powerful graph models, such as pictorial structure models [2], hierarchical models [83] and non-tree models [50, 76, 94], to represent the spatial correlations between human joints. However, these methods lack generalization ability in some cases where joints are either truncated or severely occluded.

With the availability of large-scale human pose datasets [1, 53] and the rapid development of CNNs, deep learning based methods have proven to be more robust and effective for the task of human pose estimation. Mainstream works [95, 10, 58, 7] commonly employed the multi-stage architecture to refine the output of each network stage iteratively. In particular, Wei et al. [95] proposed the Convolutional Pose Machine (CPM), which produces increasingly refined pose estimations by directly operating on belief maps from previous stages. Cao et al. [10] introduced the Part Affinity Fields (PAFs) to learn the association of body parts based on the CPM architecture, which significantly outperformed previous works. Newell et al. [58] introduced a "stacked hourglass" architecture that improves the performance by repeating bottom-up, top-down processing. This multi-stage architecture has achieved state-of-the-art results in many image-based benchmarks. Some other works [17, 98, 32] attempted to learn the feature pyramid in CNNs to capture the various spatial relationships across all scales. For example, He et al. [32] and Chen et al. [17] applied the feature pyramid structure for pose estimation by adopting the Feature Pyramid Network (FPN) [52]. Yang et al. [98] designed the Pyramid Residual Module (RPM) that learns feature pyramids using different subsampling ratios in a multi-branch network. Besides, there are also some methods [99, 86, 19, 23, 92] that combined CNNs with graphical models to learn both convolutional features and joint spatial constraints in an end-to-end network. For example, Tompson et al. [86] combined a CNN and a Markov Random Field (MRF) to exploit the spatial relationships between human joints in a unified model. Chu et al. [19] proposed a deep structured feature learning framework that models the correlations among the convolutional feature maps of body joints for accurate pose estimation. The success of all these methods demonstrates how a large spatial context is essential for CNN-based pose estimation methods.

2.2.2 Video-based Pose Estimation

Compared to pose estimation in images, estimating poses in videos is more challenging due to the complication in utilizing temporal and motion information. Early works [61, 59, 40] relying on hand-crafted features take into account temporal information through adding the temporal links between frames on the graph models. For example, Cherian et al.[18] cast the video-based pose estimation problem as an optimization problem defined on body parts with spatio-temporal links between frames.

Recent works attempt to integrate temporal cues in the advanced deep models to improve the performance of video-based pose estimation. Among them, the most common methods [67, 11, 78, 97] investigate temporal context by using optical flow. As optical flow defines the distribution of apparent velocities of movement, it can help to capture the geometric transformations between frames to refine the predicted heatmaps. For example, Song et al. [78] used optical flow to exploit image evidence from adjacent frames. Pfister et al. [67] utilized optical flow to align output heatmaps from neighboring frames to improve the performance of video pose estimation. However, optical flow requires extra data pre-processing and cannot handle large appearance variations due to person occlusions or motion blur. Some other methods [28, 54] capture the temporal dependency through LSTM, which has become a dominating tool for sequence tasks thanks to its power in long-range temporal representation. For example, Luo et al. [54] proposed a recurrent model with LSTM to consider the temporal information for pose estimation in videos. Gkioxari et al. [28] introduced a chained model using CNNs, where the pose prediction depends not only on the input but also on the output of the previous frame. There are also methods applying 3D convolution to learn representations of video clips. For example, Girdhar et al. [27] inflated the 2D convolution in the Mask R-CNN into 3D, which leverages temporal information over video clips to generate more robust pose predictions in videos. Although these methods can learn spatio-temporal representations in an end-to-end framework, they can not explicitly exploit the geometric transformation information between adjacent frames.

In our work, we propose a unified video-based pose estimation model, which explicitly explores multi-scale temporal consistency information at the feature level. Although our method is also inspired by ASPP and ConvLSTM to some degree, it focuses on effectively and efficiently capturing temporal consistency in videos. More concretely, compared with the vanilla ConvLSTM, we equip it with the deform operation to capture geometric transformations between neighboring frames at the feature level. Besides, we apply the dilated spatial pyramid module following a reduce-split-merge principle to reduce the computational cost, and we integrate it with the proposed TCE module to explore temporal consistency at multi-scale spatial levels.

2.3 Problem Formulation

Given an input video with T frames as $\{\mathbf{I}_t \in \mathbb{R}^{W \times H \times 3}\}_{t=1}^{T}$ in which $W \times H$ is the spatial size of frames, the goal of our model is to generate the corresponding sequence of human joint heatmaps $\{\mathbf{M}_t \in \mathbb{R}^{w \times h \times K}\}_{t=1}^{T}$, where $w \times h$ is the spatial size of heatmaps, and K indicates the number of joints to be estimated. Each position in the k-th channel of the heatmap corresponds to a score that indicates how much the position belongs to the k-th joint. Usually, the heatmap resolution is smaller than the input's to reduce the number of model parameters. Thus, in order to obtain the final joint positions, we select the positions with the highest score of each channel and then re-scale them to the input size. Most recent works treat the video as a sequence of independent frames. They learn a CNN to project the input frame into a convolutional feature map $\mathbf{X}_t$, and then use the Fully Convolutional Network (FCN) to predict joint heatmaps:

$$\mathbf{X}_t = \mathrm{CNN}(\mathbf{I}_t), \quad \mathbf{M}_t = \mathcal{F}_{\mathrm{FCN}}(\mathbf{X}_t). \tag{2.1}$$

These methods ignore the temporal information inherent in video data, in particular, temporal consistency between neighboring frames.

In this work, we focus on exploring temporal consistency to estimate human poses in videos. Specifically, we propose a Temporal Consistency Exploration (TCE) module that uses feature maps of the frame $\mathbf{I}_t$ and its N temporal neighboring frames as the input. By exploring the temporal consistency of adjacent frames, it associates the original feature map $\mathbf{X}_t$ with an enhanced feature map $\mathbf{H}_t$. Then, the enhanced feature map $\mathbf{H}_t$ is fed into the FCN to predict precise joint heatmaps.

$$\begin{aligned} \mathbf{H}_t &= \mathcal{F}_{\mathrm{TCE}}(\mathbf{X}_{t-N}, \ldots, \mathbf{X}_t, \ldots, \mathbf{X}_{t+N}), \\ \mathbf{M}_t &= \mathcal{F}_{\mathrm{FCN}}(\mathbf{H}_t). \end{aligned} \tag{2.2}$$

In the following sections, we first introduce a simple and effective base network for pose estimation in images. Then we introduce the details of the proposed TCE module and the multi-scale TCE module. Finally, we extend the base network with the proposed multi-scale TCE module to design a novel video-based pose estimation network.

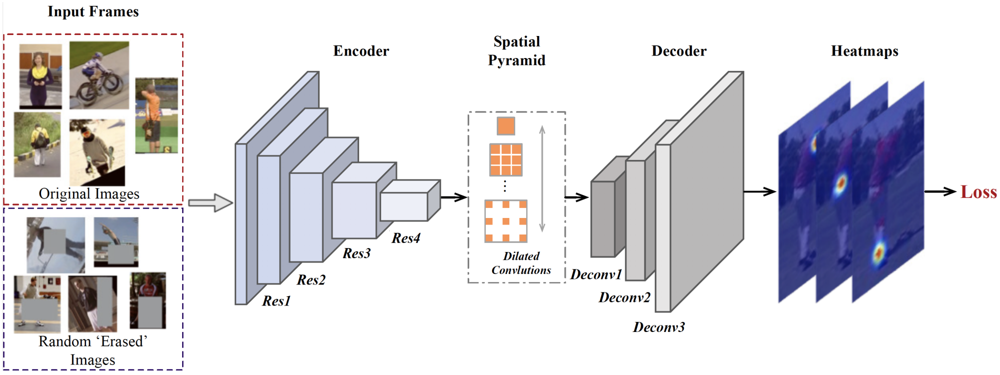


Figure 2.1 : The base pose estimation network is based on the encoder-decoder architecture, where the encoder network extracts high-level convolutional features and the decoder network recovers high-resolution heatmaps. In addition, we apply the spatial pyramid module and the random erasing technique to improve the robustness of the network.

2.4 Base Pose Estimation Network

In order to build a solid foundation for the video-based pose estimation model, we first designed a base network for estimating human poses in images. As illustrated in Figure 2.1, the network is designed based on the encoder-decoder architecture where the encoder network extracts high-level convolutional features, and the decoder network recovers the high-resolution spatial information for producing output heatmaps. For the encoder, we borrow the first four residual blocks from the Residual Network (ResNet) [34] that is a powerful CNN framework to extract high-level convolutional features. After that, the decoder adopts several deconvolutional layers to gradually enlarge the spatial dimension of the feature map. Finally, we apply a 1×1 convolutional layer to generate the output heatmap.

In our work, we use traditional encoder-decoder architecture [97], but with two significant modifications. On the one hand, to capture rich spatial context in images, a dilated spatial pyramid module is built upon the encoder. Different from the

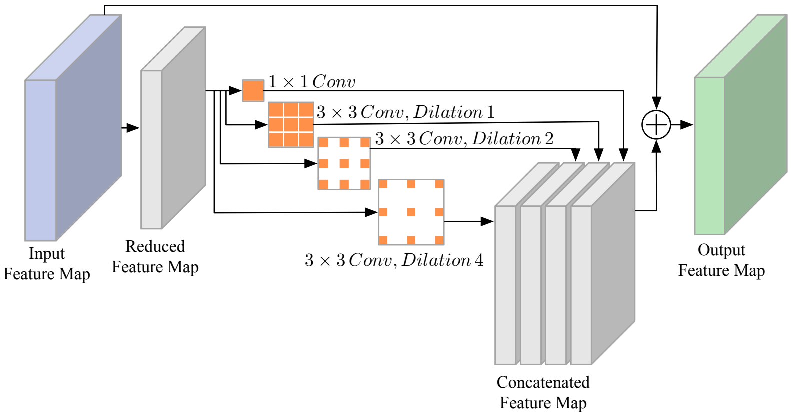

Input
Feature Map
Reduced
Feature Map
$1 \times 1\, Conv$
$3 \times 3\, Conv, Dilation\, 1$
$3 \times 3\, Conv, Dilation\, 2$
$3 \times 3\, Conv, Dilation\, 4$
Concatenated
Feature Map
Output
Feature Map

Figure 2.2 : Architecture of the dilated spatial pyramid module.

Atrous Spatial Pyramid Pooling (ASPP) [14, 77], our spatial pyramid module follows a reduce-split-merge principle for reducing computational cost. As shown in Figure 2.2,it first applies the point-wise convolution to project the high-dimensional feature map to a low-dimensional space. Then, multiple dilated convolution kernels with increasing dilation rates are parallelly adopted onto the feature map. The dilated convolution can efficiently compute convolutional features at any receptive field size without loss of resolution. Finally, the multiple outputs are concatenated and further combined with the input feature map by residual summation to produce the multi-scale representation.

We also adopt a random erasing technique inspired by Zhong et al. [108] to improve the robustness of the occlusion problem, since samples in pose estimation datasets usually exhibit limited variance in occlusion. Specifically, in the training phase, an image within a mini-batch is randomly selected to 'erase' a rectangle region of arbitrary size, and assign the pixels within the region with the mean pixel value of the dataset as shown in Figure 2.1. In this way, augmented images with various occlusion levels can be generated, and it is a simple yet effective technique for creating more robust models for the occlusion problem. Actually, other shapes or even occluding objects are also available for erasing, which has been discussed in

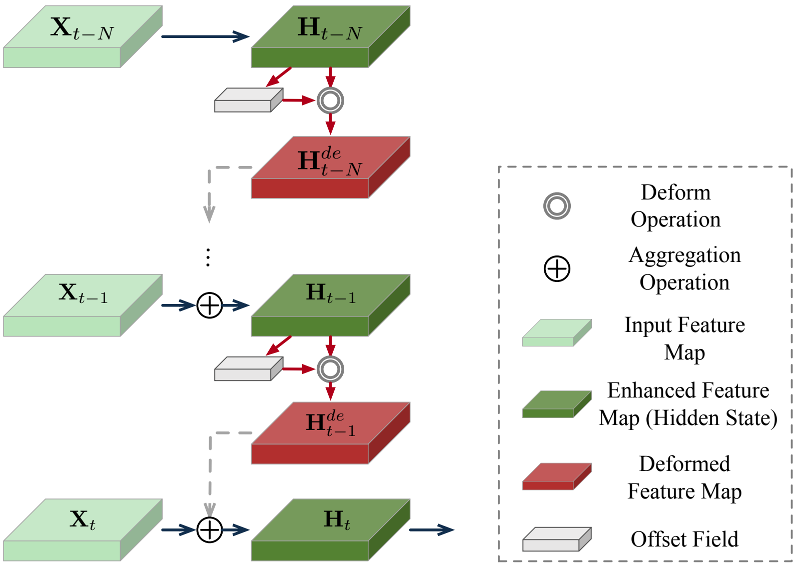


Figure 2.3 : Illustration of the TCE module. The TCE module follows a recurrent structure. It predicts the offset field to capture the geometric transformations between adjacent frames and produces the enhanced feature map through the deform and aggregation operations.

[74]. In our work, for simplifying the experiment setting, we directly use the random rectangle boxes for erasing.

2.5 Temporal Consistency Exploration

In this section, we introduce the proposed TCE module in detail. The TCE module is designed to capture temporal consistency from both temporal directions, and it processes the preceding and subsequent adjacent frames in the same way. Here, we first only consider the preceding adjacent frames to make the technique presentation clearly and briefly. Specifically, given an input frame $\mathbf{I}_t$ and its preceding temporal neighborhoods of N frames $\{\mathbf{I}_{t-N}, \dots, \mathbf{I}_{t-1}\}$, we first produce their corresponding features maps $\{\mathbf{X}_{t-N}, \dots, \mathbf{X}_t\}$ using the encoder described above. With the feature maps of neighboring frames, the TCE module produce an enhanced feature map for the target frame $\mathbf{I}_t$. The proposed TCE module follows a recurrent architecture and is formulated as:

$$\begin{aligned}\mathbf{H}_t^p &= \mathcal{T}(\mathbf{X}_t, \mathcal{A}(\mathbf{H}_{t-1})),\\ \mathbf{H}_{t-1} &= \mathcal{T}(\mathbf{X}_{t-1}, \mathcal{A}(\mathbf{H}_{t-2})),\\ &\dots,\\ \mathbf{H}_{t-N} &= \mathbf{X}_{t-N},\end{aligned} \tag{2.3}$$

where $\mathcal{T}$ refers to the aggregation operation, and $\mathcal{A}$ indicates the deform operation. $\mathbf{H}_{t-1}, \dots, \mathbf{H}_{t-N}$ represent the hidden states, and we initialize the $\mathbf{H}_{t-N}$ using the original feature map $\mathbf{X}_{t-N}$. After a series of deform and aggregation operations, as a result, we can obtain the enhanced feature map $\mathbf{H}_t^p$. The details of the TCE module are illustrated in Figure 2.3. In the following, we will introduce the deform operation and aggregation operation separately.

Deform Operation

In order to reinforce the feature map of the target frame, it is important to capture the geometric transformation between neighboring feature maps. To this end, we introduce the learnable offset field that is predicted based on the hidden state. And then, the hidden state is deformed according to the offset field for aligning it to the next feature map. In detail, we define the deform operation $\mathcal{A}$ as follows:

$$\begin{aligned}\Delta\mathbf{P} &= \mathbf{W}_{of} * \mathbf{H}_{t-1},\\ \mathbf{H}_{t-1}^{de} &= \text{Deform}(\mathbf{H}_{t-1}, \Delta\mathbf{P}),\end{aligned} \tag{2.4}$$

where $*$ refers to convolution operation. The offset field $\Delta\mathbf{P}$ is composed of the offsets of each spatial position. It is obtained by applying the convolution operation on $\mathbf{H}_{t-1}$, and $\mathbf{W}_{of}$ refers to the filter weights of the 2D convolution kernel. The offset field has the same spatial resolution with $\mathbf{H}_{t-1}$, and the channel dimension is 2 corresponding to 2-dim offset of each spatial position. With the offset field, we can obtain the deformed hidden state $\mathbf{H}_{t-1}^{de}$. For each spatial position $\mathbf{p}$ in $\mathbf{H}_{t-1}^{de}$, the value can be obtained via bilinear interpolation:

$$\begin{aligned}\mathbf{H}_{t-1}^{de}(\mathbf{p}) &= \mathbf{H}_{t-1}(\mathbf{p}+\Delta\mathbf{p}) \\ &= \sum_{\mathbf{q}} G(\mathbf{q}, \mathbf{p}+\Delta\mathbf{p}) \cdot \mathbf{H}_{t-1}(\mathbf{q}),\end{aligned} \tag{2.5}$$

where $\mathbf{q}$ enumerates all integral spatial positions on $\mathbf{H}_{t-1}$, and $G(\cdot,\cdot)$ is the bilinear interpolation operation, which can be separated into two dimensional operation as:

$$G(\mathbf{q}, \mathbf{p}) = g(q_x, p_x) \cdot g(q_y, p_y), \tag{2.6}$$

where $g(a, b) = \max(0, 1 - |a - b|)$.

Aggregation Operation

The aggregation operation can be implemented in a variety of ways. The most simple and intuitive way is through the summation operation:

$$\mathbf{H}_t^p = \mathbf{X}_t + \mathbf{H}_{t-1}^{de}. \tag{2.7}$$

To further improve the capability of the TCE module, we follow the architecture and gating mechanism of ConvLSTM, which can preserve the spatial details as well as model long-term temporal aggregation. Thus, $\mathcal{T}$ can be formulated as :

$$\begin{aligned}\mathbf{i}_t &= \sigma(\mathbf{W}_i^X * \mathbf{X}_t + \mathbf{W}_i^H * \mathcal{A}(\mathbf{H}_{t-1})) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_f^X * \mathbf{X}_t + \mathbf{W}_f^H * \mathcal{A}(\mathbf{H}_{t-1})) \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o^X * \mathbf{X}_t + \mathbf{W}_o^H * \mathcal{A}(\mathbf{H}_{t-1})) \\ \mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_c^X * \mathbf{X}_t + \mathbf{W}_c^H * \mathcal{A}(\mathbf{H}_{t-1})) \\ \mathbf{H}_t^p &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t),\end{aligned} \tag{2.8}$$

where $\mathbf{i}_t, \mathbf{f}_t, \mathbf{c}_t$ are the gates, σ and tanh are the activation function of sigmoid and hyperbolic tangent respectively. For simplicity, bias terms are omitted. '$*$' denotes the convolution operation and '$\circ$' represents Hadamard product. It is worth mentioning that the convolutional kernels for generating offset fields and output features are learned simultaneously in an end-to-end manner. This guarantees the efficiency of the proposed method.

Bidirectional Temporal Consistency Exploration

In the above, we only exploit the temporal consistency information from the preceding frame sequence. However, information from both the preceding and subsequent frames are important and complementary for predicting human joint positions. As for the original feature maps $\{\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+N}\}$ of the N subsequent frames, the TCE module processes them in the same way:

$$\begin{aligned} \mathbf{H}_t^s &= \mathcal{T}(\mathbf{X}_t, \mathcal{A}(\mathbf{H}_{t+1})), \\ \mathbf{H}_{t+1} &= \mathcal{T}(\mathbf{X}_{t+1}, \mathcal{A}(\mathbf{H}_{t+2})), \\ &\ldots, \\ \mathbf{H}_{t+N} &= \mathbf{X}_{t+N}, \end{aligned} \tag{2.9}$$

where $\mathbf{H}_{t+1}, \ldots, \mathbf{H}_{t+N}$ are the hidden states, and $\mathbf{H}_t^s$ is the enhanced feature map of the subsequent frame sequence. At last, $\mathbf{H}_t^p$ and $\mathbf{H}_t^s$ are summed up to formulate the final enhanced feature map $\mathbf{H}_t$ of the frame $\mathbf{I}_t$:

$$\mathbf{H}_t = \mathbf{H}_t^p + \mathbf{H}_t^s. \tag{2.10}$$

2.6 Multi-Scale Temporal Consistency Exploration

To capture rich spatial context in video data, we extend the spatial pyramid module described in 2.4 and designed a multi-scale TCE module that explores the geometric transformation at multi-scale spatial levels. We first apply a point-wise convolution to project the high-dimensional feature map to a low-dimensional space. Then, we simultaneously apply M dilated convolution kernels with increasing dilation rates over the feature map $\mathbf{X}_t$. After that, multi-scale feature maps are generated and fed into their respective TCE modules. Finally, $\mathbf{H}_t^\star$ that captures multi-scale spatio-temporal information is generated through concatenating the outputs of multiple TCE modules:

$$\mathbf{H}_t^\star = [\mathbf{H}_t^1, \ldots, \mathbf{H}_t^M], \tag{2.11}$$

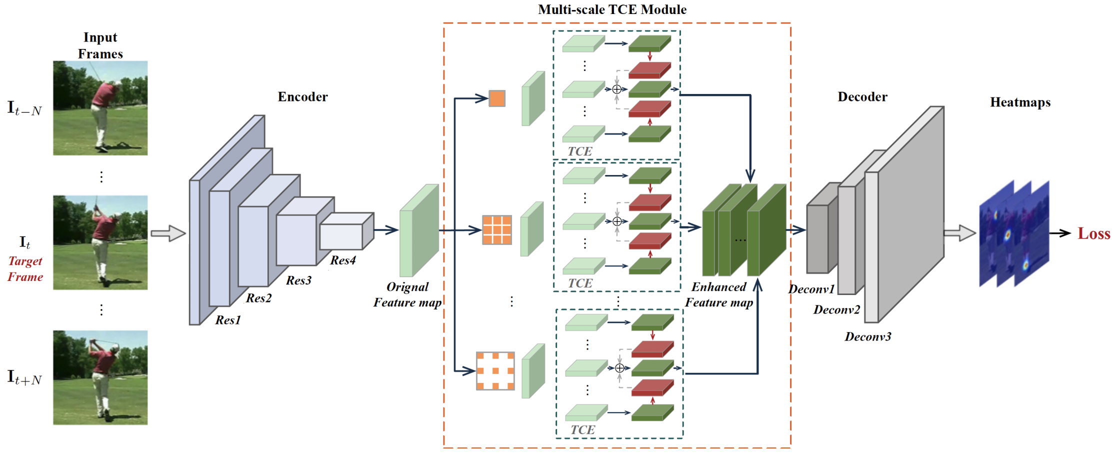


Figure 2.4 : Overall architecture of the proposed video-based pose estimation network. The proposed network is based on the encoder-decoder network architecture and extended with the multi-scale TCE module. The multi-scale TCE model fully explores the bidirectional temporal consistency information at multi-scale spatial levels. In this way, our model can generate temporally enhanced feature maps and obtain more precise human pose results.

where $[.,.]$ represents the concatenation operation, and $\{\mathbf{H}_t^n\}_{n=1}^M$ indicate the outputs of M TCE modules.

2.7 Video-based Pose Estimation Network

In this section, we introduce the video-based pose estimation network. As shown in Figure 2.4, we extend the base network with the multi-scale TCE module. The overall network architecture is similar with Peng et al. [66], which is an encoder-decoder network together with RNN-based feature refinement for face alignment. We present the details about the network architecture below.

At the bottom of the model, we use the ResNet-50 and reserve the first four residual blocks as the encoder. Given an input frame $\mathbf{I}_t \in \mathbb{R}^{256\times256\times3}$ and its N temporal neighborhoods of both directions, $\{\mathbf{I}_{t-N}, \ldots, \mathbf{I}_{t-1}\}$ and $\{\mathbf{I}_{t+1}, \ldots, \mathbf{I}_{t+N}\}$, the convolutional feature maps $\{\mathbf{X}_i \in \mathbb{R}^{8\times8\times2048}\}_{i=t-N}^{t+N}$ are first extracted through the encoder.

The multi-scale TCE module consists of $M = 4$ parallel TCE modules, which take four different scale feature maps as input. To achieve that, a point convolution is first adopted to project the feature map $\{\mathbf{X}_i\}_{i=t-N}^{t+N}$ into a low-dimensional space $\mathbb{R}^{8\times8\times512}$. Then, four convolutional kernels, including one 1×1 convolutional kernel and three 3×3 convolutional kernels with increasing dilation rates as $\{1, 2, 4\}$, are parallelly adopted to generate four different scale feature maps. For each TCE module, we equip the deform operation with 3×3 convolution kernel and produce $8 \times 8 \times 2$ offset field. The aggregation operation uses the ConvLSTM equipped with 3×3 convolutional kernels, and the hidden state of ConvLSTM cell is with the size of $8 \times 8 \times 512$. Next, the output features from four TCE modules are further concatenated to generate the enhanced feature maps $\mathbf{H}_t^{\star} \in \mathbb{R}^{8\times8\times2048}$.

The multi-scale enhanced feature map $\mathbf{H}_t^{\star}$ are then fed into the decoder. The decoder consists of three deconvolutional layers with batch normalization and ReLU activation. Each deconvolutional layer has 256 filters with 4×4 kernel (stride 2, padding 1) resulting in 2×2 up-sampling scale. Finally, a 1×1 convolutional layer is adopted to generate the output heatmaps $\mathbf{M}_t$ with spatial resolution 64×64.

Loss Function: The ground truth heatmap of the joint k of frame t, which is written as $\mathbf{M}_t^{\star k}$, is created by placing a Gaussian peak at the center location of the joint. In our work, we minimize the l_2 distance between the predicted and ground truth heatmap for each joint, and the loss function is formulated as:

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{\mathbf{p}} \|\mathbf{M}_t^{k}(\mathbf{p}) - \mathbf{M}_t^{\star k}(\mathbf{p})\|^2, \tag{2.12}$$

where $\mathbf{p}$ enumerates all integral spatial positions on the heatmap.

2.8 Results and Discussion

2.8.1 Datasets

We report our performance on two public video benchmark datasets: Sub-JHMDB [43] and Penn [106]. The Sub-JHMDB dataset has 316 video clips with all 11200 frames in the same size. It contains complete bodies with 15 joints annotated, and no invisible joint is annotated. Sub-JHMDB has three different splits of training and testing. The three splits separately have 227, 236 and 224 video clips for training and 89, 80 and 92 video clips for testing. We train our model separately and report the average result over the three splits for fair comparisons with recent methods. Besides, we also report performance on the Penn Action Dataset, which is another large video-based dataset for pose estimation. It contains in total 2326 video clips in total, with 1258 clips for training and 1068 clips for testing. 13 joints including head, shoulders, elbows, wrists, hips, knees and ankles are annotated in all frames. An additional label indicates whether a joint is visible or not in a single image. The standard evaluation protocol is only considering the visible joints. In order to further show the robustness of our method on the Penn dataset, we also perform an additional evaluation where the invisible human joints are also considered.

Even though Sub-JHMDB and Penn are large-scale video datasets, the amount of training data is still insufficient considering the high correlation among frames within the same video. Thus, to improve the generalization ability of the model, we pre-train the base network on the MPII dataset [1], which is a large image-based pose dataset. The MPII dataset consists of images taken from a wide range of human activities with a challenging array of widely articulated full-body poses, and it has around 25k images with annotations for multiple people providing 40k annotated samples (28k training, 11k testing). We pre-train our model on a subset of training images.

2.8.2 Implementation Details

Data Augmentation: Data augmentation can increase the variation of the inputs and is critical for learning robust pose estimation model. During training, we crop the frames with the target human boxes centered at images. Here, we use the person ground-truth locations provided by the datasets. Penn already annotates the bounding box within each image; the bounding boxes for sub-JHMDB are deduced from the puppet masks used for segmentation. Then, we extend either the height or the width of the human boxes to make all boxes have the same aspect ratio $(1 : 1)$. Next, we further enlarge the boxes to include additional image context by rescaling the boxes with a fixed factor 1.25. After that, boxes are randomly rotated with degree $[-40°, 40°]$, scaled with degree $[-25\%, 25\%]$ and flipped for data augmentation. Finally, all boxes are resized to a fixed resolution (256×256). Note that the transformations will be consistent for the frames within a video.

In addition to these regular data augmentation operations, random erasing is applied for improving the robustness of the model to the occlusion problem. Specifically, we set the probability of an image undergoing random erasing is 0.5. The ratio of the area of the erased rectangle region to the original image is randomly specified between $[0.02, 0.4]$, and the aspect ratio is randomly initialized between $[0.3, \frac{1}{0.3}]$.

Training Details: The training procedure of our model has two steps. In the first step, we pre-train the base network on MPII dataset. We set the batch size as 32 images, and optimize the parameters using Adam [47] algorithm. The learning rate is initialized as $1e-3$ and dropped to $1e-4$ at 90 epochs and $1e-5$ at 120 epochs. We train the image-based network for 140 epochs in total.

In the second step, we fine-tune the video-based network on the Sub-JHMDB and Penn datasets respectively. We initialize the encoder using the parameters of the pre-trained base network. And then, we fix the parameters of the encoder and

Table 2.1 : Comparisons with the state-of-the-art methods on Sub-JHMDB dataset using PCK@0.2.

Method	*Pre-train*	*Optical Flow*	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
N-best [61]	-	-	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5
ST-Part [59]	-	-	80.3	63.5	32.5	21.6	76.3	62.7	53.1	55.7
ACPS [40]	-	-	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8
Thin-Slicing [78]	-	✓	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1
LSTM PM [54]	MPII&LSP	-	98.2	96.5	89.6	86.0	98.7	95.6	90.9	93.6
Ours	-	-	97.5	97.8	88.9	85.7	98.9	94.5	90.1	93.3
Ours	MPII	-	**99.3**	**98.9**	**96.5**	**92.5**	**98.9**	**97.0**	**93.7**	**96.5**

Table 2.2 : Comparisons with the state-of-the-art methods on Penn dataset using PCK@0.2.

Method	*Pre-train*	*Optical Flow*	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
ST-Part [59]	-	-	64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0
ACPS [40]	-	-	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1
Chain [28]	-	-	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8
Thin-Slicing [78]	-	✓	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.5
LSTM PM [54]	MPII&LSP	-	98.9	98.6	96.6	96.6	98.2	98.2	97.5	97.7
Ours	-	-	99.3	98.5	97.6	97.2	98.6	98.1	97.4	98.0
Ours	MPII	-	**99.8**	**99.7**	**99.2**	**98.6**	**99.2**	**99.2**	**98.7**	**99.2**

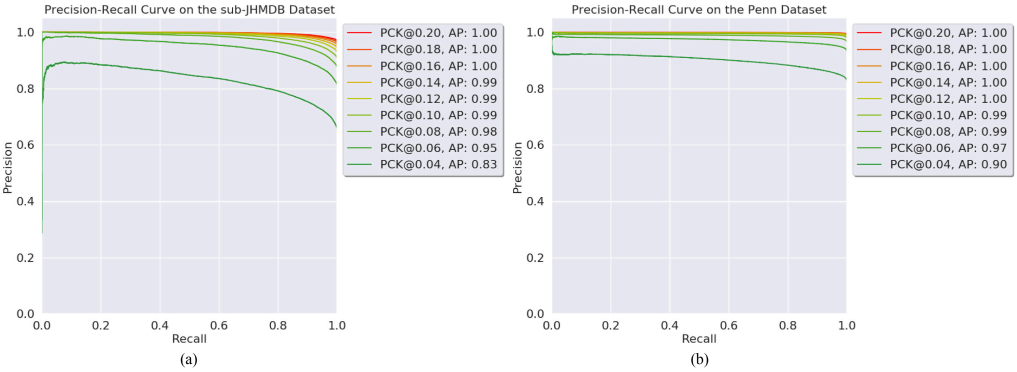
train the multi-scale TCE module and the decoder. The batch size is set to be 24 videos, and the number of neighboring frames N is set to be 6. Adam algorithm is used to optimize the network parameters. The learning rate is initialized as $1e-3$ and dropped by 10 times every 20 epochs, and there are 50 epochs in total.

Evaluation Metric: For quantitative evaluation, we adopt the PCK metric [102] to evaluate the results. An estimation is considered correct if it lies within $\alpha \cdot \max(h, w)$ from the ground truth position, where h and w refer to the height and width of the person bounding box. In our work, α is set to be 0.2 to compare with other methods consistently.

2.8.3 Performance on Video-based Pose Estimation

In this section, we compare our model with recent video-based pose estimation approaches on the Sub-JHMDB and Penn datasets. Among them, N-best [61], ST-Part [59] and ACPS [40] are conventional methods that rely on hand-crafted features. They model video temporal information through the graphical model, such as spatial-temporal And-Or graph model [59]. Thin-Slicing [78] and LSTM PM [54] are recent deep learning based models. They use the advanced CNN architecture to extract deep features of frames and adopt optical flow or LSTM to capture the temporal information in videos. Note that LSTM PM [54] pre-trained the model using the combination of two image-based datasets, MPII [1] and LSP [44].

Table 2.1 and Table 2.2 show the result comparisons. To fairly compare, we predict all joint positions, but only the visible ones are participating in the evaluation. Also, we report the results with and without pre-training on the MPII dataset respectively. Our model obtains the average accuracy of 96.4% and 99.2% on the two datasets. This is an improvement over the current supposed best performing method, LSTM PM [54], by 2.8% and 1.5% respectively. Compared with the optical flow based method [78], our model without pre-training also obtains largely accurate results. This demonstrates that our method can effectively exploit the temporal information even if optical flow is not used. When we pre-trained the model on the MPII dataset, the performance improvement on the Sub-JHMDB dataset was greater than on the Penn dataset since Sub-JHMDB is a relatively small-scale dataset. This demonstrates how pre-training the model on the image-based dataset with high diversity can avoid the risk of over-fitting on relatively small-scale video datasets and improve the generalization ability of the model. In order to further show the robustness of our method on the Penn dataset, we have performed an additional evaluation where the invisible human joints are also considered. We also perform this evaluation protocol on the most recent method, LSTM PM [23], based



(a)

(b)

Figure 2.5 : Precision-recall curves of our method on the Sub-JHMDB and Penn datasets under different PCK thresholds.

on the author's open source code. As shown in Table 2, our method also outperforms the state-of-the-art method in this evaluation protocol.

In Figure 2.5, we present the precision-recall curves of our method (with pre-training on the MPII dataset) on both datasets. We plot the precision-recall curves using different PCK thresholds α to show the effect of the threshold on the final accuracy. Figure 2.6 shows some examples of visual results in challenging settings. It shows that our model can produce accurate human poses, which demonstrates our method is robust to the problems, such as motion blur, occlusion background and scale variations.

2.8.4 Ablation Study

In this section, we present a detailed ablation analysis to show the effectiveness of our model.

Analysis of the TCE Module

In order to evaluate the effect of the proposed TCE module, we design a baseline and several variants to compare their performance on the split 1 of the Sub-JHMDB dataset. The baseline and variants we designed are listed as follows:

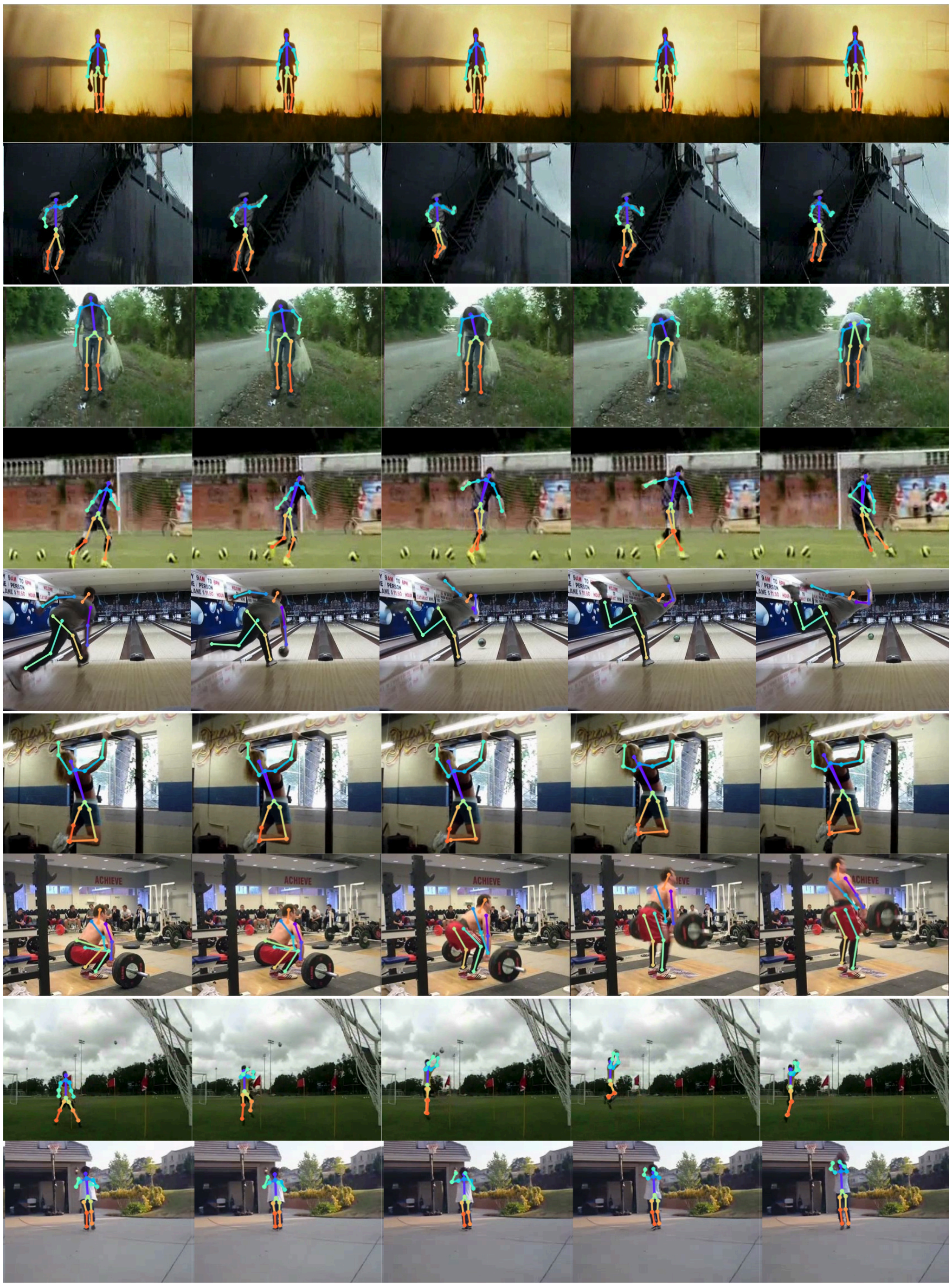


Figure 2.6 : Examples of pose estimation results on the Sub-JHMDB and Penn datasets. (row 1,2,3,4,5) Results of challenging samples (i.e., occlusion background and motion blur); (row 6,7,8,9) Results of persons with significant scale variations. Zoom-in for details.

Table 2.3 : PCK@0.2 of different variants on the split 1 of Sub-JHMDB dataset.

Model	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
Res50	98.2	96.0	91.2	88.0	98.5	95.0	93.2	94.0
Res50-OF	99.1	97.6	92.8	88.7	98.5	95.5	93.3	94.8
Res50-TCE-S	98.8	97.2	92.5	89.9	98.5	96.4	94.4	95.1
Res50-TCE-C	99.4	97.9	94.1	91.0	98.7	96.8	94.0	95.7
Res50-TCE-BC	99.3	97.9	94.8	91.7	98.8	97.0	94.2	96.0

- **Res50:** This is the baseline network that considers the video as independent frames. It is based on the encoder-decoder network described in 2.4 and uses the ResNet-50 as the encoder.

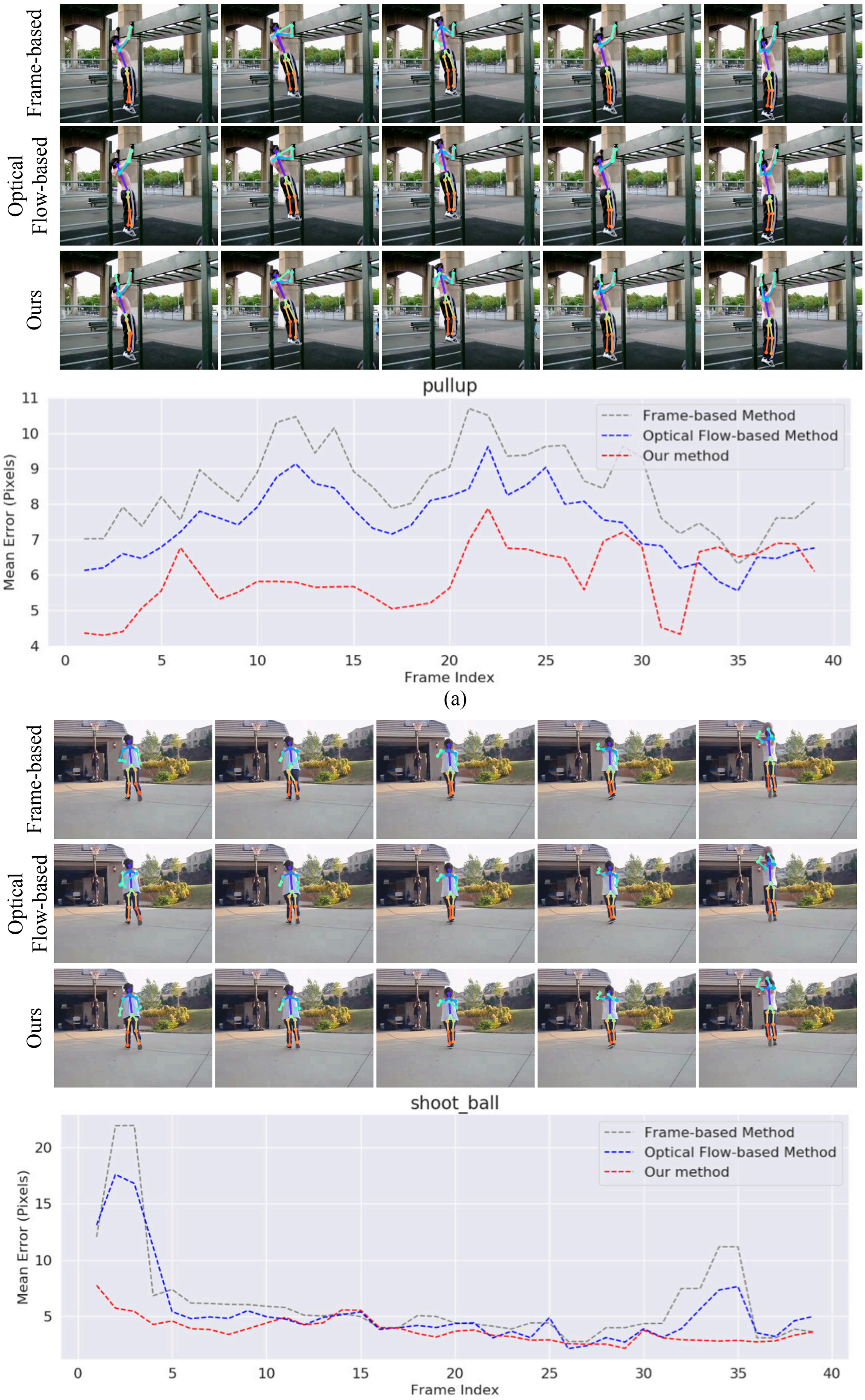
- **Res50-OF:** This variant is designed for comparing with the method using optical flow post-processing. Here, we use Flownet v2.0 [37] to extract the backward and forward flow of input videos. Then, we use the technique proposed by Pfister et al. [67] to warp the heatmaps of the neighboring frames, and average them with the heatmap of the target frame to get the final heatmap.

- **Res50-TCE-S:** In this variant, we adopt the proposed TCE module and use the basic summation operation as the aggregation operation.

- **Res50-TCE-C:** This variant adopts ConvLSTM for temporal aggregation.

- **Res50-TCE-BC:** This variant applies bidirectional ConvLSTM to consider temporal consistency information from both directions.

We initialize them using the parameters of the pre-trained base network and separately train them on the split 1 of Sub-JHMD dataset. Table 2.3 illustrates the results. We can observe that even the baseline network can achieve state-of-the-art results thanks to the solid foundation of the base network. As a post-processing method, Res50-OF achieves 0.8% improvements compared with the baseline. Over-



(a)

(b)

Figure 2.7 : Quantitative analysis of the smoothness of our results.

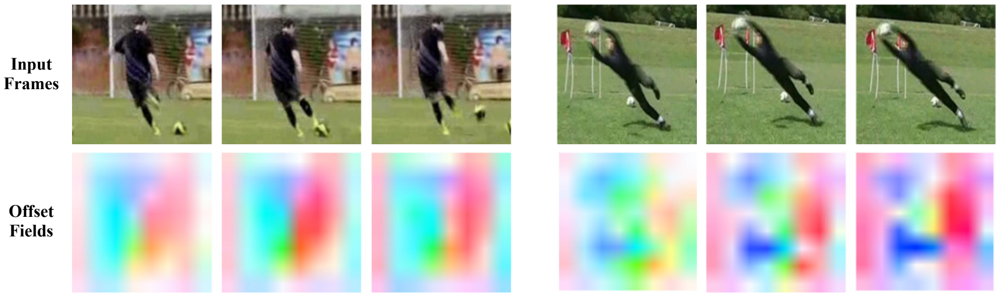


Figure 2.8 : Visualization of the predicted offset fields.

all, the variants equipped with the TCE module achieve more significant improvements and outperform the Res50-OF. Res50-TCE-S outperforms the baseline by 1.1%, and Res50-TCE-C equipped with ConvLSTM aggregation function achieves better results by 1.7%. This illustrates how the proposed TCE module is necessary, which effectively exploits geometric transformations between feature maps of neighboring frames. The performance of Res50-TCE-BC is further improved and outperforms the baseline by 2.0% due to the exploration of bidirectional temporal consistency information.

To validate that our method can obtain smooth results, we present a quantitative analysis in Figure 2.7. We present the mean error (distance from ground-truth in pixels) curves over time of two action categories (pull up and shoot ball). It shows that our method (Res50-TCE-BC) significantly reduces the joint position errors compared with the frame-based method and optical flow-based method, which improves the prediction stability over frames. Also, we visualize two examples of results obtained by the three kinds of methods. We can observe that our method obtains apparent improvements especially for the joints with severe occlusion (i.e., elbow and hand). As a comparison, the optical flow-based method has limited improvements.

In Figure 2.8, we visualize two examples of the predicted offset field ΔP using

Table 2.4 : PCK@0.2 of variants with different loss functions on the split1 of the Sub-JHMDB dataset.

Model	Regression Loss	Integral Loss	Heatmap Loss
Res50	92.0	92.8	94.0
Res50-TCE-BC	93.8	95.4	96.0

the technique [4]. We found that the offset field ΔP is not directly correlated with the optical flow, and it cannot explicitly indicate the pixel-level motion information as the optical flow does. This is because the predicted offset field is based on high-level convolutional feature maps, where each position on the feature map represents a response value of a large receptive field of the image. Moreover, the TCE module is trained without explicit alignment between video frames. Thus, the TCE module can not guarantee that the predicted offset filed has the same semantics as the image-level optical flow.

Analysis of the Loss Functions

Table 2.4 presents a comparison of different loss functions. It compares the performance of the variants Res50 and Res50-TCE-BC using three different kinds of loss functions: heatmap loss, regression loss, and integral loss. Here, heatmap loss is what we are using in our model. As for the regression loss, we replace the decoder with a fully connected layer to predict joint coordinates and calculate the $L1$ distances between predicted joints and ground-truth ones. Integral loss, which is proposed by Sun et al. [81], adopts the soft-argmax upon predicted heatmaps to convert them into joint coordinates in a differentiable way, and then calculates the joint location loss as supervisions. As shown in Table 4, the heatmap loss is overall best-performing, and Res50-TCE-BC achieves significant improvement in every loss

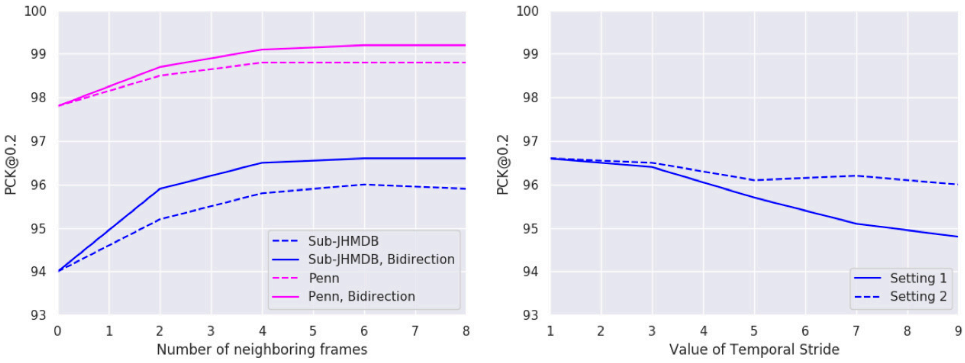
Table 2.5 : Analysis of the random erasing and spatial pyramid on the split1 of the Sub-JHMDB dataset using PCK@0.2.

Model	*Random Erasing*	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
Res50-TCE-BC	-	99.2	97.7	93.6	90.1	98.5	96.3	93.7	95.3
	✓	99.3	97.9	94.8	91.7	98.8	97.0	94.2	96.0
Res50-SS-TCE-BC	-	98.9	98.2	93.4	89.5	99.1	96.8	93.9	95.4
	✓	99.4	98.5	95.6	91.4	98.8	97.5	93.9	96.2
Res50-MS-TCE-BC	-	99.1	97.7	94.2	90.9	97.9	97.0	93.7	95.6
	✓	99.3	98.9	96.5	92.5	98.9	97.0	93.7	96.5

function compared with Res50. This shows that the TCE module works for different prediction strategies and loss functions of 2D pose estimation.

Analysis of the Spatial Pyramid and Random Erasing Techniques

Here, we evaluate the effectiveness of the techniques, spatial pyramid and random erasing, used in our work. First, we design a variant named Res50-MS-TCE-BC that is equipped with the multi-scale TCE, and compare the performance between Res50-MS-TCE-BC and Res50-TCE-BC on the split 1 of the Sub-JHMDB dataset. Compared with the TCE module, the multi-scale TCE module increases the number of channels of the feature map fed into the decoder. To validate the performance improvement is indeed caused by the spatial pyramid, we design another variant named Res50-SS-TCE-BC. Different from the Res50-MS-TCE-BC, this variant applies the same convolutional kernel (3×3 convolutional kernel with dilation rate 1) for the four parallel TCE modules. Furthermore, we apply different data augmentation strategies (using random erasing or not) to train all variants for analyzing the random erasing technique. As the results shown in Table 2.5, Res50-MS-TCE-BC achieves consistently better performance than Res50-TCE-BC and Res50-SS-TCE-



(a)

(b)

Figure 2.9 : Plots of the results with different numbers of neighboring frames and temporal stride values.

BC. This proves that the fusion of spatial pyramid and the TCE module can further boost the performance of pose estimation in videos. Besides, the models that are trained equipped with random erasing can obtain higher accuracy, especially for joints that are easily obscured like Elbow and Wrist. This illustrates that random erasing technique can effectively improve the robustness of the model for the occlusion problem.

Analysis of the Number of Neighboring Frames and Temporal Stride

Our model uses the target frame and the neighboring frames as its input. In this section, we explore the effect of the number of neighboring frames by training the model with different N, i.e., $N = 0, 2, 4, 6, 8$. Moreover, we train the model under two strategies: one only considers the preceding frames, the other considers both preceding and subsequent frames. The experiment results on the split 1 of Sub-JHMDB dataset and Penn dataset are shown in Figure 2.9 (a). It is obvious that bidirectional temporal consistency modeling helps to achieve better performance. Next, when $N = 0$, the performance drops a lot since no temporal information is considered. When N increases to around 4, the performance improves and the rate of rising gradually decreases. At last, the accuracy remains stable when N increases to 6. On the one hand, this illustrates that more neighboring frames can provide more sufficient temporal information, which helps to predict the offset field from the hidden state, and as a result, produces better enhanced feature maps. On the other hand, it shows the frames that are far from the target frame have relatively limited effects.

The default value of the temporal stride used in the above part is 1. To analyze the influence of using different temporal strides, we consider bi-directional neighboring frames with $N = 6$ and set different temporal stride values TS, i.e., $TS = 1, 3, 5, 7, 9$. Here, we run experiments on the split 1 of the Sub-JHMDB dataset and consider two different settings. The first is training the network with $TS = 1$ and testing it using different temporal strides. The second is using the same temporal stride during training and testing. The results of the two settings are shown in Figure 2.9 (b). We can observe that the performance drops as the temporal stride value becomes larger in the first setting. Thus the temporal stride during training and testing should be identical. In the second setting, the curve has small fluctuations, which illustrates the temporal stride has a limited effect on the final performance. Besides, the network obtains the best performance when the temporal stride sets 1, which shows that the TCE module can better capture the temporal consistency when the temporal stride is small.

2.8.5 Runtime Analysis

In this section, we present the runtime analysis of our model. On the one hand, our model is a unified framework and does not need any pre- or post-processing. This guarantees that the speed of our model is faster than the optical flow based method such as Thin-Slicing [78], because optical-flow based methods require extra computation of optical flow calculation. Here, we present the speed of common optical flow estimation methods for reference: LDOF [9] takes about 49.64s per frame, Flownet v2.0 [37] takes about 50ms per frame. On the other hand, we compare our model with the latest video-based method, LSTM PM, under the same configuration. The results show that LSTM Pose Machine achieves processing speed of about 25 ms per-frame, and our method achieves around 28ms per-frame. It means our model achieves a significant performance improvement at the expense of a little speed.

2.9 Conclusion

In this work, we have presented a unified deep network for estimating human poses in videos. To efficiently explore the temporal consistency in videos, we proposed the novel TCE module that captures geometric transformations between frames at the feature level. On the basis of the TCE module, we further integrated it with the spatial pyramid to explore time consistency at multi-scale spatial levels. Finally, we designed a video-based pose estimation network by extending the encoder-decoder architecture with the multi-scale TCE module. The experimental results showed that our model achieves better performance than recent video-based approaches on two popular video datasets. Moreover, we showed the effectiveness and efficiency of our model through a detailed ablative analysis.

Chapter 3

Geometry-driven Self-supervised Method for 3D Human Pose Estimation

3.1 Introduction

3D human pose estimation has attracted substantial interest for its vast potential on various applications including human-computer interaction, virtual reality and action recognition. With the great success of deep learning, many researchers [55, 65] applied the neural network to predict 3D human poses from monocular images. Estimating 3D poses using neural networks mainly faces two main challenges. First, a typical neural network model needs a large amount of training data. 3D pose annotations are collected through the marker-based Motion Capture (MoCap) system, which is an expensive process. Secondly, there are well-founded geometrical theories on how to project 2D images to 3D skeletons. Simply using a neural network to approximate this projection may lead to the network subject to overfitting training data.

To alleviate the above challenges, weakly/self-supervised learning paradigms have been increasingly explored in recent works [72, 96, 13, 71]. Re-projection loss [89], which does not require explicit 3D ground-truth, has become a commonly used technique. It re-projects estimated 3D poses back to the 2D space and calculates the loss between the input and re-projected 2D poses as supervision. However, due to the depth ambiguity problem where multiple 3D body configurations can explain the same 2D projection, the re-projection loss cannot yield accurate and realistic 3D poses. For example, as shown in Figure 3.1, since re-projection loss only con-

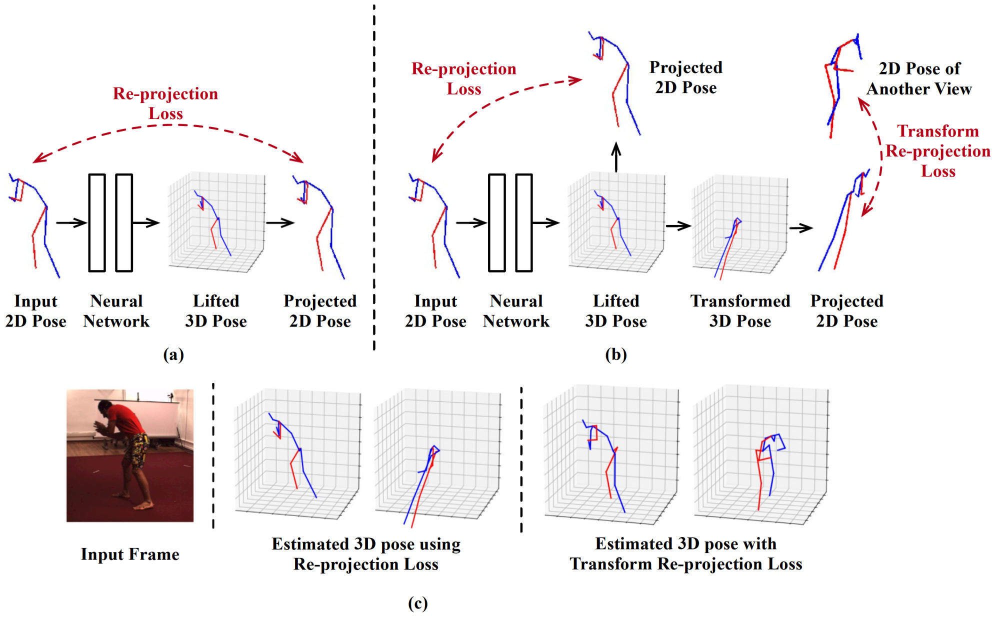


Figure 3.1 : (a) The network trained with re-projection loss. (b) The proposed network trained with transform re-projection loss. (c) The comparisons of results estimated by the above two networks. The results are shown in two different views.

strains the estimated 3D pose at a specific camera angle, it may result in an invalid human pose when observed from another angle. Although some techniques such as adversarial loss [89, 90] and kinematic constraints [30, 65], have been proposed to constrain the estimated 3D poses into a semantic sub-space, they usually require some extra unpaired 3D pose annotations (without 2D-3D correspondence) to make the network memorize the distribution of real 3D skeletons.

3D pose datasets [38, 56] are usually collected under the configuration with multiple calibrated cameras. The consistency information between multiple camera views has not been fully explored in recent weakly/self-supervised methods. Although the recently proposed work [48] has explored multi-view geometry to train a network, they utilized triangulation on detected multi-view 2D poses to generate ‘ground-truth’ 3D poses, which are subsequently used to train a 3D pose network. However, this naive application of 3D multi-view geometry is sub-optimal due to the noises introduced in the 2D pose detection at each individual camera. The detected 2D poses are combined to produce its 3D pose, which may further produce noisy supervision signals. Besides, the process of generating pseudo ground-truth is redundant.

In this paper, we propose a novel self-supervised approach to take advantage of the geometric prior for training a 3D pose estimation model. We formulate 3D pose estimation as 2D keypoint estimation followed by 2D-to-3D pose lifting. The first stage is compatible with any state-of-the-art 2D keypoint detector, and our work concentrates on training the 2D-to-3D lifting network without using any additional 3D ground-truth data. Specifically, in order to overcome the depth ambiguity problem, we design the transform re-projection loss. As shown in Figure 3.1(b), it transforms the lifted 3D poses from current view to another randomly selected view through rigid transformation, and then calculates the re-projection loss between the transformed 3D pose and the 2D pose of the target view. As a result, it can effectively constrain the estimated 3D poses by considering multi-view consistency. Due to the self-occlusion problem, some 2D joints may be invisible at the frame of a particular camera angle, which may lead to inaccurate 2D keypoint detections. However, they may be visible from other camera angles. Thus, the same human joint will obtain different 2D detection confidences on different camera views. We acquire the confidence weights from estimated 2D keypoint heatmaps and use them to integrate losses of different camera views, which makes our method more robust to noisy 2D detections. Finally, we introduce a root position regression branch to restore the global 3D poses during training. In this way, we can reserve the scale information of re-projected 2D poses, which can improve the accuracy of the predicted 3D poses. Moreover, in order to train the root position branch and lifting branch simultaneously from scratch, we propose a pre-training technique to help the network converge.

We perform extensive experiments on two popular 3D human pose datasets: Human3.6M [38] and MPI-INF-3DHP [56]. The results demonstrate that our method achieves state-of-the-art performance. The contributions of our work are summarized as follows:

- We propose a self-supervised approach to train the 2D-to-3D lifting network without any 3D pose annotations. It only relies on geometry knowledge to construct supervision signals, which leads to a better generalization ability.
- We design the transform re-projection loss, which is an effective technique to exploit multi-view consistency information and constrain the estimated 3D poses during training. Moreover, we integrate it with the 2D joint confidences of different camera views to alleviate the self-occlusion problem.
- The proposed method achieves state-of-the-art results on two popular 3D pose benchmarks compared with recent weakly/self-supervised methods.

3.2 Related Work

3.2.1 3D Human Pose Estimation

3D human pose estimation is a long-standing problem and has been considerably studied in the past few years. Recently, following the great success of deep learning, modern 3D human pose estimation techniques are usually formulated as learning-based frameworks. These works can be generally classified into two categories. The first class of methods [82, 63, 56, 30, 81] directly predict the depth from monocular images through the deep convolutional neural networks (DCNNs). The second category [39, 23, 84, 55, 23, 12] is the two-stage pipeline, which first obtains 2D joint locations through the advanced 2D keypoint detector such as Stacked Hourglass (SH) network [58] and Cascaded Pyramid Network (CPN) [17], and then lifts them into 3D space. In order to learn the mapping between 2D and 3D joint positions, various 2D-to-3D lifting network backbones were designed. For example, Martinez et al. [55] proposed a simple baseline using a simple neural network with only two fully connected layers, while achieved surprising results. Since human skeletons are with the graph-like structure, several works [107] also attempted to exploit the novel Graph Convolution Networks (GCNs) to capture the semantic relationships between human joints for accurate 3D human pose regression. Besides, there are also works [65, 3, 110, 109] that considers temporal information from frame sequence to produce more robust predictions. In our work, we follow the two-stage pipeline. Moreover the proposed approach is compatible with any recent 2D-to-3D lifting network backbone.

3.2.2 Weakly/Self-supervised Approaches

Recently, weakly/self-supervised approaches have received much attention due to the difficulty of gathering 3D pose annotations. In order to train the network without explicit 3D pose annotations, the prior of camera projection geometry was commonly explored, and some geometry-driven methods were proposed. Among them, re-projection loss is one of the most widely used technique [46, 96, 90, 65, 93, 8]. However, using re-projection loss alone cannot accurately constrain the depth of skeletons due to the depth ambiguity problem. Some works [30, 65, 72] alleviated this problem by enforcing the bone length similarity between predicted and ground-truth skeletons. Adversarial loss [100, 89, 90, 46] is another popular technique to regularize the predicted 3D poses. It encourages the output 3D poses on the real human manifold by introducing a real/fake 3D skeleton discriminator. For example, [89] proposed the Adversarial Inverse Graphical Network (AIGN), which uses the adversarial prior to match the distribution between the predictions and a collection ground-truth for the task such as 2D-to-3D lifting and image-to-image translation. [90] proposed a weakly-supervised method with the adversarial super-

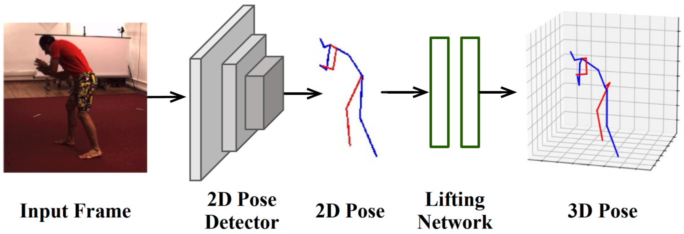


Figure 3.2 : The overall architecture of our method that follows a two-stage pipeline.

vision for 3D human pose estimation. [13] exploits the geometric self-consistency of the lift-reproject-lift process with the adversarial prior of 2D poses. As we analyzed above, the bone length constraint and adversarial loss still require unpaired 3D pose annotations for counting the bone length or training the real/fake 3D skeleton discriminator. Differently, we proposed a self-supervised approach that solely relies on the camera geometry prior, which can result in better generalization ability.

3.3 Overview of the Method

Overall, the proposed method follows a two-stage pipeline, as shown in Figure 4.1. First, we use the state-of-the-art 2D pose estimation network to predict 2D poses from input frames. Here, we denote $\mathbf{X} \in \mathbb{R}^{N\times 2}$ as N detected 2D joint locations. Meanwhile, we obtain their corresponding confidence scores $\mathbf{w} \in \mathbb{R}^{N}$ from estimated keypoint heatmaps through the *max* operation. With the detected 2D poses, we learn a neural network $\mathcal{N}$ to project them into 3D space. Similarly, we define $\mathbf{Y} \in \mathbb{R}^{N\times 3}$ as the output 3D joint locations. Following the protocol with previous works, we estimate zero-centered 3d poses where the values of $\mathbf{Y}$ are the 3D positions relative to the fixed root joint (pelvis).

The architecture of the lifting network $\mathcal{N}$ is designed inspired by [55]. The input layer takes the concatenated coordinates of N human joints and applies a fully connected layer with 1024 output channels. Then it is followed by four blocks that are

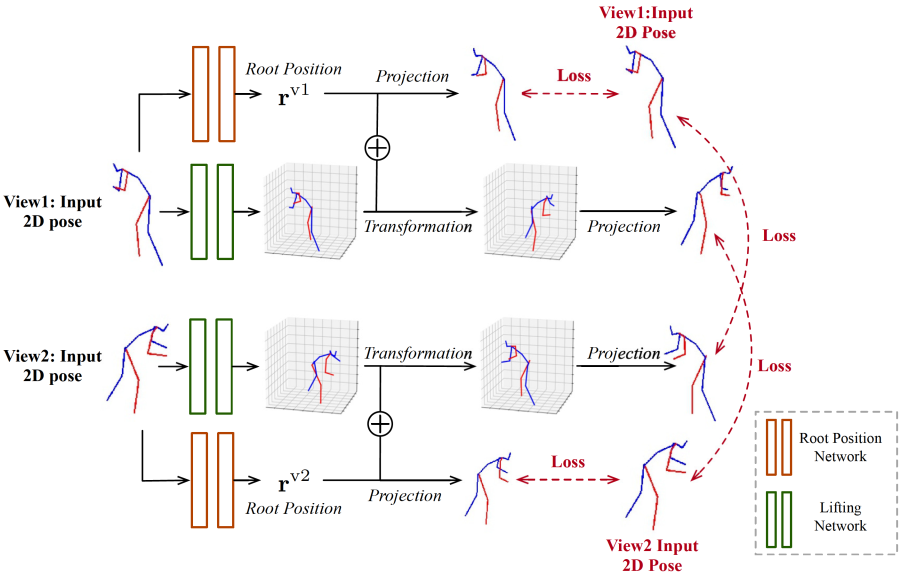


Figure 3.3 : The architecture of the proposed self-supervised training approach.

surrounded by residual connections. For each block, several fully connected layers (1024 channels) followed by Batch Normalization, rectified linear units, and dropout, are stacked for efficiently mapping the 2D pose features to high-level features. Finally, the features extracted by the last residual block are fed into an extra linear layer ($N \times 3$ channels) to output 3D poses.

3.4 Self-supervised Approach

In this section, we introduce the proposed self-supervised approach for training the lifting network. The training process takes as input the detected 2D poses of a pair of frames that are captured from two different views at the same time. With the paired frames, we first detect their 2D poses $\mathbf{X}^{\mathrm{v}1}$ and $\mathbf{X}^{\mathrm{v}2}$ and their corresponding confidence weights of each joint $\mathbf{w}^{\mathrm{v}1}$ and $\mathbf{w}^{\mathrm{v}2}$. Then, we feed the 2D poses into the lifting network and obtain their estimated 3D poses $\mathbf{Y}^{\mathrm{v}1}$ and $\mathbf{Y}^{\mathrm{v}2}$.

3.4.1 Two-branch Training Architecture

For training the lifting network without 3D ground-truth annotations, we design the transform re-projection loss. It involves the perspective projection and view transformation operations, which require global 3D joint positions. Without global 3D joint positions, we can not obtain the absolute depth of the person in the camera coordinate, which results in unknown scale when re-projecting 3D poses back to 2D space. Existing methods commonly normalize the scale of 2D skeletons to overcome the scale ambiguity problem. However, it must be used in conjunction with the kinematic constraint or adversarial loss to output realistic 3D poses.

In our work, we design another branch, named root position branch, to help train the lifting network. It predicts root joint positions, $\mathbf{r}^{\mathrm{v}1}$ and $\mathbf{r}^{\mathrm{v}2}$, which are added to relative 3D poses predicted by the lifting network to restore global 3D poses, $\tilde{\mathbf{Y}}^{\mathrm{v}1}$ and $\tilde{\mathbf{Y}}^{\mathrm{v}2}$. The root position network has the same architecture with the lifting network, and they do not share any weights. The two branches can be optimized simultaneously using multi-view consistency information, and the loss function and detailed training procedure will be discussed in the following sections.

3.4.2 Loss Function

With the global 3D poses, we first re-project them back to the 2D space following the perspective projection ρ.

$$\begin{aligned}\rho(\tilde{\mathbf{Y}}_i^{\mathrm{v}1}) &= \begin{bmatrix} f_x^{\mathrm{v}1}\tilde{\mathbf{Y}}_i^{\mathrm{v}1}(x)/\tilde{\mathbf{Y}}_i^{\mathrm{v}1}(z) + c_x^{\mathrm{v}1} \\ f_y^{\mathrm{v}1}\tilde{\mathbf{Y}}_i^{\mathrm{v}1}(y)/\tilde{\mathbf{Y}}_i^{\mathrm{v}1}(z) + c_y^{\mathrm{v}1} \end{bmatrix}, \\ \rho(\tilde{\mathbf{Y}}_i^{\mathrm{v}2}) &= \begin{bmatrix} f_x^{\mathrm{v}2}\tilde{\mathbf{Y}}_i^{\mathrm{v}2}(x)/\tilde{\mathbf{Y}}_i^{\mathrm{v}2}(z) + c_x^{\mathrm{v}2} \\ f_y^{\mathrm{v}2}\tilde{\mathbf{Y}}_i^{\mathrm{v}2}(y)/\tilde{\mathbf{Y}}_i^{\mathrm{v}2}(z) + c_y^{\mathrm{v}2} \end{bmatrix},\end{aligned} \tag{3.1}$$

where f_x and f_y refer to the focal lengths, c_x and c_y define the principal points, $\tilde{\mathbf{Y}}_i^{\mathrm{v}1}(x)$ indicates the value of x coordinate of i^{th} joint position of $\tilde{\mathbf{Y}}^{\mathrm{v}1}$. And then, we calculate the l_2 loss between the input and re-projected 2D poses as supervisions,

$$\begin{aligned}\mathcal{L}_{\text{reproj}} = & \sum_{i}^{N} \mathbf{w}_i^{\text{v1}} \|\mathbf{X}_i^{\text{v1}} - \rho(\tilde{\mathbf{Y}}_i^{\text{v1}})\|^2 \\ & + \mathbf{w}_i^{\text{v2}} \|\mathbf{X}_i^{\text{v2}} - \rho(\tilde{\mathbf{Y}}_i^{\text{v2}})\|^2,\end{aligned} \tag{3.2}$$

where $\mathbf{w}_i^{\text{v1}}$ and $\mathbf{w}_i^{\text{v2}}$ are the confidence scores of the i^{th} joints of two views. Here, we use the confidence scores of detected 2D poses to integrate the re-projection loss of different views. The view with smaller 2D confidence value makes less contribution to the loss value, which reduces the impact of the noisy 2D detections for the lifting network training.

However, simply using the re-projection consistency will encounter the depth ambiguity problem. To overcome the problem, we design the transform re-projection loss, which constrains the predicted 3D skeletons from multiple perspectives. Specifically, we transform the estimated 3D pose from one view to another through the rigid transformation τ as follows:

$$\begin{aligned}\tau(\tilde{\mathbf{Y}}_i^{\text{v1}}) &= \mathbf{R}_{\text{1to2}} \left(\tilde{\mathbf{Y}}_i^{\text{v1}} - \mathbf{t}_{\text{1to2}} \right), \\ \tau(\tilde{\mathbf{Y}}_i^{\text{v2}}) &= \mathbf{R}_{\text{2to1}} \left(\tilde{\mathbf{Y}}_i^{\text{v2}} - \mathbf{t}_{\text{2to1}} \right),\end{aligned} \tag{3.3}$$

where $\mathbf{R}_{\text{1to2}}, \mathbf{R}_{\text{2to1}} \in \mathbb{R}^{3\times3}$ are the rotation matrixes, and $\mathbf{t}_{\text{1to2}}, \mathbf{t}_{\text{2to1}} \in \mathbb{R}^3$ are the transformation vectors. With the extrinsic parameters of two cameras $\mathbf{R}_1$, $\mathbf{t}_1$ and $\mathbf{R}_2$, $\mathbf{t}_2$, we can directly obtain the rigid transformation parameters,

$$\begin{aligned}\mathbf{R}_{\text{1to2}} &= \mathbf{R}_2 \mathbf{R}_1^{\text{T}}; \quad \mathbf{t}_{\text{1to2}} = \mathbf{R}_1 \left(\mathbf{t}_2 - \mathbf{t}_1 \right), \\ \mathbf{R}_{\text{2to1}} &= \mathbf{R}_1 \mathbf{R}_2^{\text{T}}; \quad \mathbf{t}_{\text{2to1}} = \mathbf{R}_2 \left(\mathbf{t}_1 - \mathbf{t}_2 \right).\end{aligned} \tag{3.4}$$

If extrinsic parameters of cameras do not exist, we can use the positions of 2D joins of two views as calibration targets [48]. We assume the first camera as the center of the coordinate system, which means $\mathbf{R}_1$ is an identity matrix and $\mathbf{t}_1$ is a zero vector. For corresponding joints in $\mathbf{X}^{\text{v1}}$ and $\mathbf{X}^{\text{v2}}$, we find the fundamental matrix $\mathbf{F}$ satisfying $\mathbf{X}_i^{\text{v1}} \mathbf{F} \mathbf{X}_i^{\text{v2}} = 0, i = 1 \ldots N$, using RANSAC algorithm. From $\mathbf{F}$, we calculate the

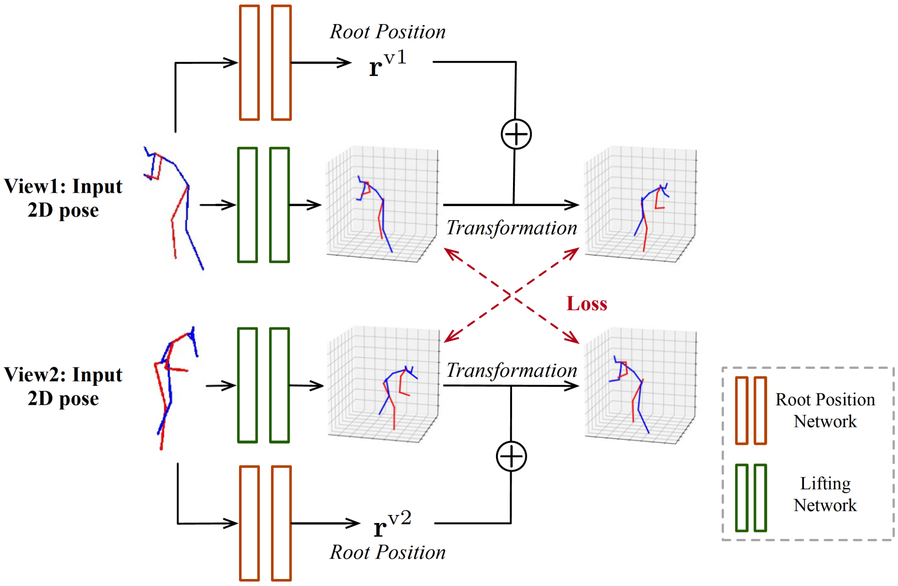


Figure 3.4 : Illustration of the model pre-training.

essential matrix $\mathbf{E}$ by $\mathbf{E} = \mathbf{P}_{\mathrm{v2}}^{\mathrm{T}}\mathbf{F}\mathbf{P}_{\mathrm{v1}}$, where $\mathbf{P}_{\mathrm{v1}}$ and $\mathbf{P}_{\mathrm{v2}}$ are the projection matrixes of cameras. By decomposing $\mathbf{E}$ with SVD, we obtain four possible solutions to $\mathbf{R}_{\mathrm{1to2}}$ and $\mathbf{t}_{\mathrm{1to2}}$. We decide on the correct one by verifying possible pose hypotheses using cheirality check. In the similar way, we can get $\mathbf{R}_{\mathrm{2to1}}$ and $\mathbf{t}_{\mathrm{2to1}}$. Since the calibrated $\mathbf{t}_{\mathrm{1to2}}$ and $\mathbf{t}_{\mathrm{2to1}}$ are unit vectors, we need to multiply them by the distance between two camera centers.

Next, according to multi-view consistency that the 2D projection of the transformed 3D skeleton should be the same with the 2D input of the target view, we design the transform re-projection loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{t-reproj}} = \sum_{i}^{N} & \mathbf{w}_i^{\mathrm{v1}} \|\mathbf{X}_i^{\mathrm{v2}} - \rho(\tau(\tilde{\mathbf{Y}}_i^{\mathrm{v1}}))\|^2 \\ & + \mathbf{w}_i^{\mathrm{v2}} \|\mathbf{X}_i^{\mathrm{v1}} - \rho(\tau(\tilde{\mathbf{Y}}_i^{\mathrm{v2}}))\|^2.\end{aligned} \tag{3.5}$$

In this way, we construct supervision signals entirely relying on camera geometric prior. Compared with existing techniques that require unpaired 3D pose annotations or kinematic constraints, the proposed approach is simple and effective.

3.5 Training

It is challenging to train two inter-dependent branches from scratch without ground-truth annotations. We find that the network cannot converge if we train it with random initialization. Thus, we design a pre-training technique to warm-up the network. As shown in Figure 3.4, the pre-training loss can be formulated as:

$$\mathcal{L}_{\text{pre-train}} = \sum_{i}^{N} \|\tau(\tilde{\mathbf{Y}}_i^{\text{v1}}) - \tilde{\mathbf{Y}}_i^{\text{v2}}\|^2 + \|\tau(\tilde{\mathbf{Y}}_i^{\text{v2}}) - \tilde{\mathbf{Y}}_i^{\text{v1}}\|^2. \tag{3.6}$$

It is designed according to multi-view consistency that the transformed 3D pose and the estimated 3D pose of the target view should be the same. Although this loss is not able to guide the lifting network to produce valid 3D poses, it effectively regularizes the output space of the root position branch. It can be regarded as an advanced initialization of the root position branch, which greatly reduces the difficulty of network convergence.

After pre-training, the network is fine-tuned using the re-projection loss and transform re-projection loss,

$$\mathcal{L}_T = \mathcal{L}_{\text{reproj}} + \lambda \mathcal{L}_{\text{t-reproj}}, \tag{3.7}$$

where λ is a hyper-parameter that is adapted to set under different datasets. We set λ as 1 and 1.5 respectively in Human3.6M and MPI-INF-3DHP datasets.

3.6 Experiments

3.6.1 Datasets

We perform extensive evaluations on two publicly available benchmarks.

- Human3.6M (H36M) [38] is one of the largest datasets for 3D human pose estimation, which is captured by MoCap system. It consists of 3.6 million images with 11 actors performing 15 actions such as eating, sitting and walking. They are captured from 4 calibrated cameras with known intrinsic and extrinsic parameters. In our experiments, we follow the standard protocol with 17-joint subset, use subjects S1, S5, S6, S7, S8 for training and S9, S11 for testing.

- MPI-INF-3DHP (3DHP) [56] is a recently proposed 3D pose dataset constructed with both constrained indoor scenes and complex outdoor scenes. We use the five chest-height cameras and the provided 17 joints (compatible with H36M) for training, and we use the official test set, which contains 2929 frames from six subjects performing seven actions, for evaluation.

Evaluation Metrics: For the H36M dataset, we consider two popular evaluation protocols. **Protocol 1** is the Mean Per Joint Position Error (MPJPE) in millimeters (mm). MPJPE is the mean euclidean distance between the ground-truth and predicted positions of the joints. **Protocol 2** is the Procrustes MPJPE (P-MPJPE), which aligns the estimated 3D pose to the ground-truth by a rigid transformation called Procrustes Analysis before computing the MPJPE.

The evaluation metrics for the 3DHP dataset include the adapted Percentage of Correct Keypoints (PCK) and corresponding Area Under Curve (AUC) [56]. The PCK indicates the percentage of joints whose estimated position is within 15cm of the ground-truth.

Data Augmentation: The H36M dataset has only four calibrated camera views. Training with more camera views can improve model performance and generalization ability. We follow the technique proposed by [23] to simulate a series of virtual camera views. We extend the H36M dataset from 4 views to 12 views containing 8 virtual camera views, and we obtain the corresponding 2D pose of each sample through perspective projection to augment the training set. The detailed analysis will be shown in the following sections.

3.6.2 Implementation Details

In order to enable the proposed two-branch network to converge without any explicit 3D pose supervision, the training procedure contains two stages. First, we pre-train the network using the $\mathcal{L}_{\text{pre-train}}$ loss. We use the Adam as the optimizer and train the network for 20 epoches with learning rate 0.001. Next, the network is trained using the $\mathcal{L}_T$ loss for 300 epoches. The learning rate starts from 0.001 and drops by 0.1 each 100 epoches. During evaluation, for consistency with other works, we only use the 2D-to-3D lifting branch to predict the relative 3D poses in the camera space, and not use the root position branch. We implement our method using the deep learning toolbox Pytorch.

3.6.3 Ablation Study

Analysis of Transform Re-projection Loss

In order to evaluate the effectiveness of the proposed transform re-projection loss, we compare it with the existing popular technique, adversarial loss. We design several variants and compare the results under Protocol #1 (MPJPE) and Protocol #2 (P-MPJPE) on the H36M dataset. All variants use 2D poses extracted by the CPN network as inputs. Table 3.1 presents the quantitative results, and Figure 3.5 shows the results of different variants on several hard samples, i.e., with serious self-occlusion or far from the camera. It is obvious that only using the re-projection loss will obtain strange 3D skeletons that do not conform the human kinematics. Although adversarial loss can constrain the 3D poses using unpaired 3D pose annotations, it still can not produce precise 3D poses, especially when encountering samples with serious self-occlusion. Compared with adversarial loss, our method achieve significant performance improvements, and the MPJPE and P-MPJPE decrease by 47.6 and 32.7 (mm). This shows that the transform re-projection loss can effectively help the network learn geometric knowledge, which further constrains the estimated 3D poses to get more accurate results.

3.6.4 Analysis of Data Augmentation

The data augmentation can resolve the depth ambiguity problem to some extent by introducing more camera views. Specifically, the MPJPE and P-MPJPE will decrease by extra 2.0 and 1.6 (mm) when using data augmentation. This verifies that training with more camera views can effectively facilitate the model performance.

Table 3.1 : Detailed results on H36M dataset under Protocol #1 and Protocol #2.

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Reproj	390.2	441.6	479.3	422.8	503.4	479.0	400.6	471.5	568.5	662.2	483.6	423.8	473.2	414.5	413.5	468.5
Reproj+ADV	81.7	93.0	99.3	97.3	106.8	134.7	81.8	101.0	113.2	151.2	100.7	97.0	121.3	111.6	108.3	106.6
Trans_Reproj	49.7	54.5	58.0	56.8	63.4	80.0	52.4	52.7	71.4	78.3	58.9	55.2	60.0	43.8	49.6	59.0
Trans_Reproj+DA	48.7	53.6	54.7	55.1	61.3	76.1	51.5	50.3	68.0	75.9	56.7	53.8	58.8	42.6	47.9	57.0
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Reproj	147.1	148.0	174.7	153.1	165.5	162.7	176.1	136.2	156.5	192.5	230.1	147.6	150.5	160.0	154.6	163.7
Reproj+ADV	64.8	69.4	77.4	74.4	78.4	94.2	60.4	68.9	81.5	113.1	74.3	70.4	84.8	82.4	81.6	78.4
Trans_Reproj	39.6	42.6	45.7	46.0	47.6	57.1	41.0	39.2	55.4	59.9	46.4	42.5	47.1	34.4	41.0	45.7
Trans_Reproj+DA	38.2	41.3	43.5	44.4	45.4	54.7	39.3	38.0	53.2	59.2	45.0	40.7	46.2	33.0	39.4	44.1

[1] ADV refers to the adversarial loss; DA means that the network is trained with data augmentation.

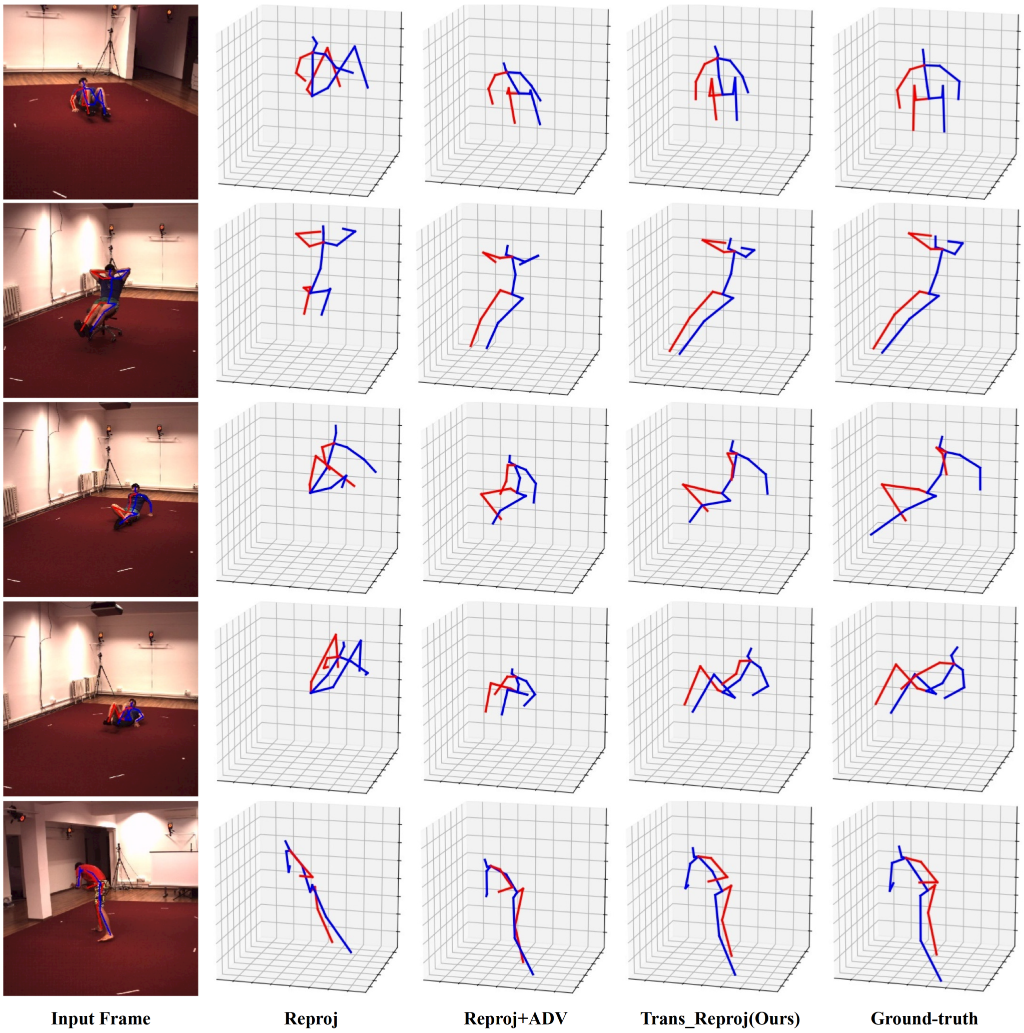


Figure 3.5 : Results of different variants in some hard examples.

Analysis of Backbones

Our method does not depend on any particular backbone. In this part, we investigate the performance of our method with different 2D-to-3D network backbones. ResLinear [55] is the earliest and most commonly used backbone, which consists of fully connected layers and residual connections. TemporalDilated [65] is the latest proposed backbone that can explore the temporal information using dilated temporal convolutions. We feed it with 243 neighboring frames as inputs during training and testing. As shown in Table 3.2, our approach can achieve competitive results when

Table 3.2 : Comparisons of different backbones on the H36M datasets.

Backbone	Protocol 1	Protocol 2
ResLinear	59.7	45.0
Ours	57.0	44.1
TemporalDilated	56.1	43.2

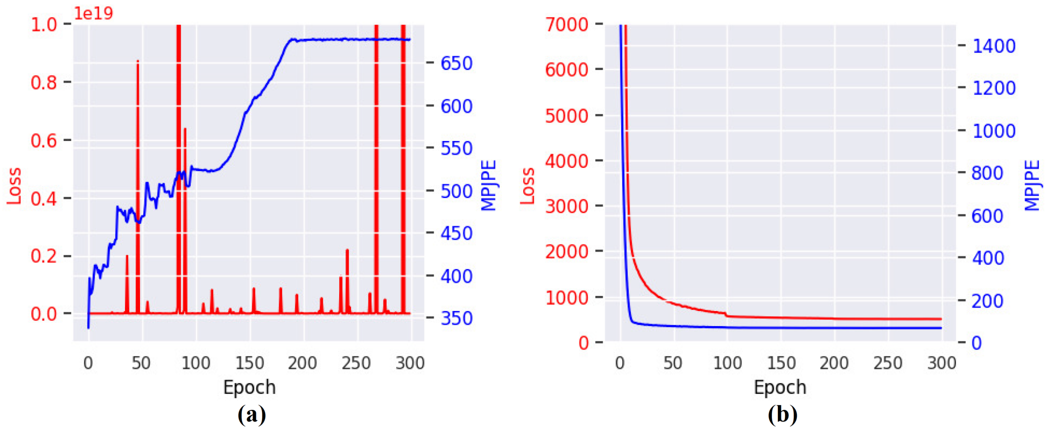


Figure 3.6 : (a) and (b) are the loss and MPJPE curves of the network trained without and with pre-training respectively.

using the simple ResLinear backbone. Therefore, the improvements of our method are not merely due to the better backbone. When using the TemporalDilated, our method gains obvious improvements, which benefits from the exploration of temporal information. These results illustrate that the proposed self-supervised training technique has strong versatility and is suitable for any novel 2D-to-3D network architecture.

Analysis of Network Pre-training

In this section, we show the effectiveness of the network pre-training. Since the proposed network is trained without any 3D pose annotation, the pre-training is very important for our two-branch networks. As shown in Figure 3.6, the loss and MPJPE curves without pre-training violently oscillate. The network fails to converge despite our best efforts at tuning the hyper-parameters. In contrast, the loss curve rapidly decreases, and we achieve low MPJPE value when using the proposed pre-training technique. It illustrates that pre-training technique is vital and effective in our work.

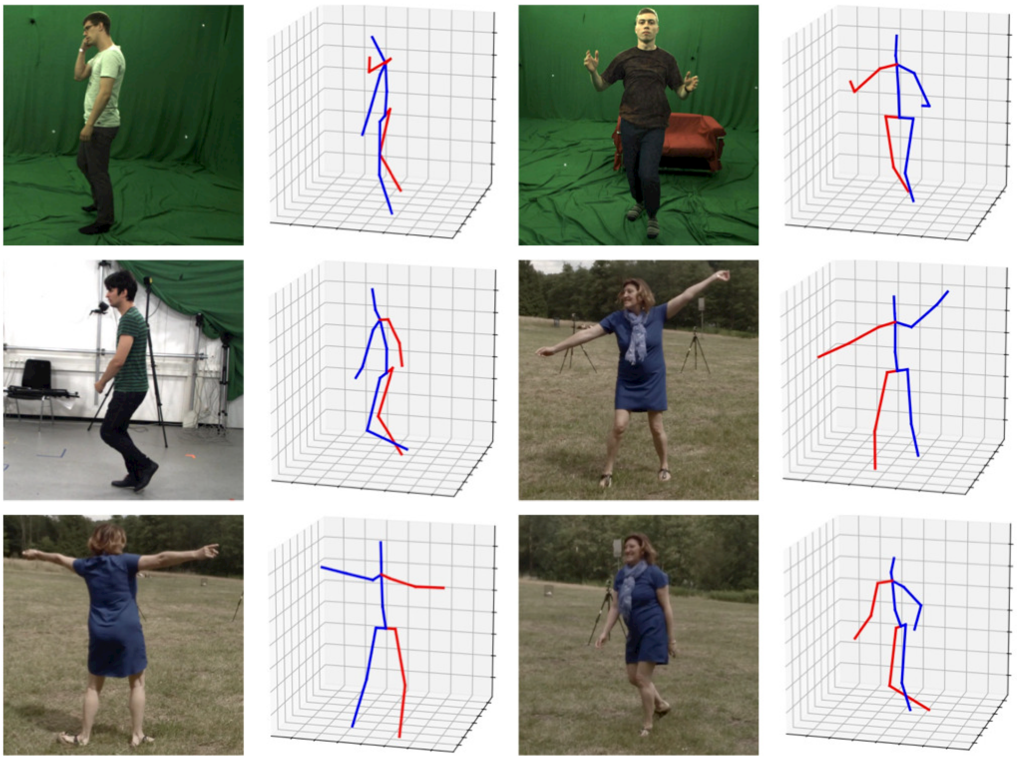


Figure 3.7 : Quantitative results of our method (trained on the H36M dataset) on the 3DHP dataset.

Analysis of Generalization Ability

To demonstrate the generalization ability of our model, we train the network on the H36M dataset and evaluate it on the test split of the 3DHP dataset, which includes challenging outdoor scenes. We present some examples in Figure 3.7. It shows that our approach can successfully recover 3D poses on the datasets without being trained on them.

Table 3.3 : Comparisons with recent weakly/self-supervised methods on the H36M dataset under evaluation Protocol #1 and Protocol #2.

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	**Avg**
Pavlakos *et al.* CVPR'17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	118.4
Tung *et al.* ICCV'17 (†)	77.6	91.4	89.9	88	107.3	110.1	75.9	107.5	124.2	137.8	102.2	90.3	-	78.6	-	97.2
Wandt *et al.* CVPR'19 (†)	77.5	85.2	82.7	93.8	93.9	101.0	82.9	102.6	100.5	125.8	88.0	84.8	72.6	78.8	79.0	89.9
Wang *et al.* PAMI'19 (⋆;†)	50.0	60.0	54.7	56.6	65.7	52.7	54.8	85.9	118.0	62.5	79.6	59.6	41.5	65.2	48.5	63.7
Kocabas *et al.* CVPR'19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76.6
Ours	49.7	54.5	58.0	56.8	63.4	80.0	52.4	52.7	71.4	78.3	58.9	55.2	60.0	43.8	49.6	59.0
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Zhou *et al.* ICCV'17 (†)	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.1	66.0	5.4	63.2	55.3	64.9
Drover *et al.* ECCV'18	60.2	60.7	59.2	65.1	65.5	63.8	59.4	59.4	69.1	88.0	64.8	60.8	64.9	63.9	65.2	64.6
Rhodin *et al.* ECCV'18 (†)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	98.2
Wandt *et al.* CVPR'19 (†)	53.0	58.3	59.6	66.5	72.8	71.0	56.7	69.6	78.3	95.2	66.6	58.5	63.2	57.5	49.9	65.1
Kocabas *et al.* CVPR'19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.5
Chen *et al.* CVPR'19 (⋆)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	68.0
Ours	39.6	42.6	45.7	46.0	47.6	57.1	41.0	39.2	55.4	59.9	46.4	42.5	47.1	34.4	41.0	45.7

[1] (⋆) denotes that it takes advantage of the temporal information; (†) denotes that it requires unpaired or part of 3D pose annotations.

3.6.5 Comparisons with State-of-the-art Methods

In this section, we compare our method with recent weakly/self-supervised methods. First, we compare with them on the H36M dataset using protocol #1 and protocol #2 in Table 4.3. Tung et al. [89], Wandt et al. [90], and Zhou et al.[111] are based on re-projection loss and require additional unpaired 3D pose annotations. Compared with them, our method has an explicit improvement and obtains average errors of 59.0mm and 45.7mm under two evaluation protocols. Our method also outperforms Kocabas et al. [48] that adopts multi-view information. We present its result obtained in the case of using ground-truth extrinsic parameters for fair comparisons. It illustrates that the proposed approach is a more effective way to exploit multi-view information. Table 4.4 shows the comparisons with state-of-the-art methods on the 3DHP dataset. In this setting, we train the network on the train set of the 3DHP dataset, and evaluate it on the test set following the PCK and AUC metrics. As seen, the PCK and AUC of our method reach 74.1 and 41.4 respectively, which outperform previous methods.

Table 3.4 : Comparisons with recent weakly/self-supervised methods on the 3DHP dataset.

Method	PCK	AUC
Zhou *et al.* ICCV'17	69.2	32.5
Kocabas *et al.* CVPR'19	64.7	-
Chen *et al.* CVPR'19	71.1	36.3
Ours	74.1	41.4

3.7 Conclusion

In this work, we proposed a new self-supervised approach for 3D human pose estimation. The approach explored multi-view consistency to construct supervision signals for training a 2D-to-3D lifting network, which can effectively overcome the depth ambiguity problem. Note that our method simply applied multi-view information during training, and required only single view inputs during inference. Meanwhile, we designed a two-branch training architecture and pre-training technique to ensure the network can successfully converge and achieve excellent performance. Extensive ablation studies on the H36M and 3DHP datasets illustrated the effectiveness and generalization ability of our approach. The experiment results showed that our method obtained a superior performance over recent weakly/self-supervised methods.

Chapter 4

Self-supervised Method for 3D Human Pose Estimation with Consistent Factorization Network

4.1 Introduction

3D human pose estimation has recently received considerable attention by the computer vision community, due to its vast potential on various applications, including human-computer interaction, virtual reality, and human action recognition. In particular, with the promise of deep learning methodologies, many researchers [55, 65] start to adopt neural networks to predict 3D human poses from monocular images. However, learning 3D human pose estimation networks is bottlenecked by the lack of large 3D annotated datasets. Thus, weakly/self-supervised learning paradigm became a practical alternative in recent times.

Most of the works [100, 13] directly regress the depth values or 3D joint positions at the corresponding 2D camera view, thanks to the powerful fitting capability of deep neural networks. However, estimating 3D joint positions from monocular images is an ill-posed problem where there exist multiple 3D human poses for a single 2D skeleton. This problem is particularly prominent for weakly/self-supervised methods that use the re-projection loss [111]. To overcome this challenge, the GAN-based technique, such as adversarial loss [100], has been proposed, and it becomes the most commonly used to constraint the predictions and make it appear to be “human-like”. It does so by encouraging the generated 3D poses to map onto the real-human manifold using a real/fake 3D skeleton discriminator. However, the adversarial loss usually requires extra unpaired 3D pose annotations (without 2D-3D correspondence) to train the real/fake classifier.

On the other hand, some other works [89, 90, 92] represent the 3D human pose as a linear combination of a dictionary of view-agnostic 3D pose basis. They design the pose estimation network to predict the coefficients associated with the dictionary and the camera viewpoint, instead of predicting the 3D joints directly. This technique reduces the space of all allowed 3D coordinates and makes the predictions within the "human" sub-space. However, previous methods [89, 23] commonly learn the dictionary using off-the-shelf dimension reduction techniques, such as PCA or the sparse coding. These techniques are generic and only focus on the statistical aspect of the data, but ignore the geometric aspect specific to 3D human pose variations. Moreover, since these methods cannot guarantee the 3D shape and viewpoint are to become fully disentangled, they still require the adversarial loss as an additional constraint to overcome the projection ambiguity.

In our work, we propose a novel self-supervised framework to systematically address the above problems. In a nutshell, our method learns a neural network to consistently factorize the 3D shape and camera viewpoint from the input 2D skeleton. Consistent factorization means that 2D projections from different viewpoints of the same 3D skeleton should have the same canonical reconstruction, and be distinguished completely by camera viewpoints. Based on this intuitive fact, we design an effective technique to learn the factorization network, which takes advantage of multi-view information to constrain the canonical reconstruction. Similar to [89], we also represent the 3D human pose as a combination of a dictionary of pose basis. However, in order to reconstruct robust 3D human poses, we exploit the underlying 3D geometry of human pose and learn the dictionary from 2D poses through Non-Rigid Structure from Motion (NRSfM), which is a classical technique to reconstruct deformable 3D shapes from monocular scenes. Besides, we follow the prior assumption that 3D shapes are compressible via multi-layer sparse coding [49], and use the hierarchical dictionary to reconstruct 3D human poses. Specifically, to obtain the hierarchical dictionary, we optimize an encoder-decoder network, where neural network weights serve as the dictionary, through minimizing the NRSfM objective. It is an efficient way and does not require 3D pose annotations for learning the hierarchical dictionary. Consequently, our method can effectively learn a 3D human pose estimation network and overcome the projection ambiguity problem with the hierarchical dictionary and consistent factorization network.

We have performed extensive experiments on two popular 3D human pose datasets: Human3.6M [38] and MPI-INF-3DHP [56], and the results show that our method achieves state-of-the-art performance. Finally, we summarize the contributions of our work as follows:

- We propose a novel self-supervised method for 3D human pose estimation, which effectively overcomes the projection ambiguity problem by consistently factorizing the 3D human shape and camera viewpoint using multi-view information.
- In order to reconstruct robust canonical 3D human poses, we exploit the underlying 3D geometry of human poses to learn a hierarchical dictionary from 2D poses. It is an effective way and does not require extra 3D pose annotations.
- The proposed method achieves state-of-the-art results on two popular 3D pose benchmarks compared with recent weakly/self-supervised methods.

4.2 Related Work

4.2.1 3D Human Pose Estimation

3D human pose estimation has been considerably studied in the past few years. With the great success of deep learning, recent researches mainly focus on the learning-based framework. Here, we classify these methods into two categories. Some of them [82, 63, 56, 30, 81] predict 3D poses directly from single images through the deep convolutional neural networks (CNNs). For example, Sun et al. [81] proposed the integral regression approach, which directly regresses 3D joint coordinates through a CNN. Pavlakos et al. [63] trained a CNN to predict the voxel likelihoods for each human joint through a fine discretization of the 3D space. The other methods [39, 23, 84, 55, 12] follow the two-stage framework. They first estimate 2D joint positions through an advanced 2D keypoint detector [95, 58, 17], and then lift them into 3D space. Various 2D-to-3D lifting networks have been designed to learn the mapping between 2D and 3D joint positions. For example, Martinez et al. [55] adopted a neural network with only two fully connected layers while achieving surprising results. It has become the most commonly used baseline. Several works [107] exploited the novel Graph Convolution Networks (GCNs) to capture the graph-structure of the human skeleton for accurate 3D human pose regression. Besides, there are also works [65, 3, 110, 109] that exploit temporal information in videos to generate more smooth results.

4.2.2 Weakly/Self-supervised Approaches

Recently, weakly/self-supervised approaches have received much attention due to the lack of large 3D human pose datasets In order to train the network without explicit 3D pose annotations, re-projection loss has been proposed and become a common technique [8, 46, 96, 90, 65, 93]. However, re-projection loss faces the projection ambiguity problem where multiple 3D human poses can explain the same 2D skeleton. In order to alleviate this problem, several kinds of techniques have been proposed. First, the most intuitive way [30, 65, 72] is to enforce the bone length similarity between predicted and real human skeletons, which makes the predicted 3D poses conform the human kinematics. Second, the adversarial loss [100, 89, 90, 46], which is inspired by the recent Generative Adversarial Network (GAN), is the most commonly used technique. It encourages the output 3D poses on the real human manifold through a real/fake 3D skeleton classifier. For example, Tung et al. [89] designed the Adversarial Inverse Graphical Network (AIGN) for the task such as 2D-to-3D lifting and image-to-image translation. They adopted the adversarial loss to match the distribution between the predictions and a collection ground-truth. Chen et al. [13] exploited the geometric self-consistency of the lift-reproject-lift process, as well as added a 2D pose discriminator to output valid 3D poses. Wandt et al. [90] proposed the RepNet that learns a mapping from a distribution of 2D poses to 3D poses using the adversarial supervision. The above two techniques still require extra unpaired 3D pose annotations to constrain estimated 3D human poses. The third category is geometry-based [16, 103], which introduces multi-view information for constraining predicted 3D human poses. For example, Rhodin et al. [71] pre-trained an encoder-decoder network that predicts an image from one view to another. In this way, the network learned geometry-aware representations and then was fine-tuned using a small amount of supervision to predict 3D human poses. Kocabas et al. [48] applied Epipolar geometry to obtain 3D poses from estimated 2D joint positions of multiple views, and used them for training a 3D pose estimator. These methods usually explicitly require multi-view inputs as well as the corresponding extrinsic parameters (rotation matrix or 6-DoF camera extrinsic) that are calculated through camera calibration or Epipolar geometry. Otherwise, they still require a 2D/3D pose discriminator as an additional constraint.

4.2.3 Dictionary-based Approaches

Although most of recent methods directly regress depth values or 3D positions of human joints at the corresponding camera view, some methods advocate using a dictionary of 3D pose basis elements to represent 3D human skeleton. For example, Tung et al. [89] adopt PCA on the orientation-aligned training set to obtain a shape dictionary, and represent a 3D human pose as a linear combination of 3D shape basis. Novotny et al. [60] also represent 3D human poses as a combination of shape basis. They consider the dictionary as the weights of a linear layer and learn it with the 3D pose estimation network in an end-to-end manner. Most recently, some researches make advances in Non-rigid Structure from Motion (NRSfM), which is a classical technique to reconstruct 3D shapes of articulated 3D points. For example, Kong and Lucey [49] proposed the Deep-NRSfM network architecture as solving a multi-layer block sparse dictionary learning problem, and achieve high-quality 3D human pose reconstructions. Wang et al. [91] proposed a new knowledge distilling algorithm applicable to Deep-NRSfM based on dictionary learning, and achieves weakly-supervised learning using solely 2D human joint annotations. In our work, we take advantage of the recent advances in the dictionary learning technique for NRSfM, and focus on fully disentangling 3D human shape and camera viewpoint from input 2D poses.

4.3 Problem Formulation

Here, we start by giving a brief problem formulation. We then introduce our main contributions of the consistent factorization network in Section 4.4 and hierarchical dictionary in Section 4.5.

Given an image, we denote $\mathbf{X} \in \mathbb{R}^{P\times 2}$ as the 2D coordinates of P human joints of the target person. The target of our method is to predict the corresponding 3D human pose, and we define $\mathbf{Y} \in \mathbb{R}^{P\times 3}$ as the 3D human pose under the camera coordinate. In our work, we decompose $\mathbf{Y}$ as the canonical 3D human pose and camera viewpoint, which is formulated as:

$$\mathbf{Y} = \hat{\mathbf{Y}}\mathbf{R}, \quad \hat{\mathbf{Y}} = [\mathbf{D}\varphi]_{P\times 3}, \tag{4.1}$$

where $\hat{\mathbf{Y}}$ is the 3D human pose under the canonical viewpoint, which is represented as the combination of a code vector $\varphi \in \mathbb{R}^K$ and a dictionary of 3D pose basis $\mathbf{D} \in \mathbb{R}^{3P \times K}$. Operator $[]_{P\times 3}$ is defined as reshaping the vectorized 3D pose into matrix form with dimension $P \times 3$. As for the camera viewpoint, it is represented by a rotation matrix $\mathbf{R} \in SO(3)$. In order to avoid calculating the orthonormal constraint, we predict the exponential coordinate of the rotation matrix $\omega \in \mathbb{R}^3$, and $\mathbf{R}$ can be obtained by the *Rodrigues' rotation formula* $\mathbf{R} = \text{expm}[\omega]_\times$, where expm is the matrix exponential and $[.]_\times$ is the hat operator.

However, traditional dictionary learning approaches usually cannot guarantee sufficient freedom on 3D human pose variations. In our work, we follow the prior assumption that 3D shapes are compressible via multi-layer sparse coding [49], and define that the canonical 3D human pose satisfies:

$$\begin{aligned} \hat{Y} &= \mathbf{D}_1\varphi_1, \quad \|\varphi_1\|_1 \leq \lambda_1, \varphi_1 \geq 0, \\ \varphi_1 &= \mathbf{D}_2\varphi_2, \quad \|\varphi_2\|_1 \leq \lambda_2, \varphi_2 \geq 0, \\ &\vdots \quad , \quad \vdots \\ \varphi_{n-1} &= \mathbf{D}_n\varphi_n, \quad \|\varphi_n\|_1 \leq \lambda_n, \varphi_n \geq 0, \end{aligned} \tag{4.2}$$

where $\hat{Y}$ is the vectorization of $\hat{\mathbf{Y}}$, $\mathbf{D}_1 \in \mathbb{R}^{3P\times K_1}, \mathbf{D}_2 \in \mathbb{R}^{K_1\times K_2}, \ldots, \mathbf{D}_n \in \mathbb{R}^{K_{n-1}\times K_n}$ are the hierarchical dictionary, and $\varphi_i \in \mathbb{R}^{K_i}$ are multi-layer sparse code vectors that are constrained to be sparse and non-negative. In this prior, each 3D human pose is represented by a hierarchical dictionary and corresponding sparse codes. Compared with the single-level dictionary, codes obtained through the hierarchical dictionary not only minimize the reconstruction error at their individual levels, but is also regularized by the codes from other levels. This helps to impose more constraints on code recovery while having stronger 3D pose expressibility.

Consequently, with the hierarchical dictionary, we can reconstruct the 3D human pose under the camera coordinate through the 3D pose code φ_n and exponential co-

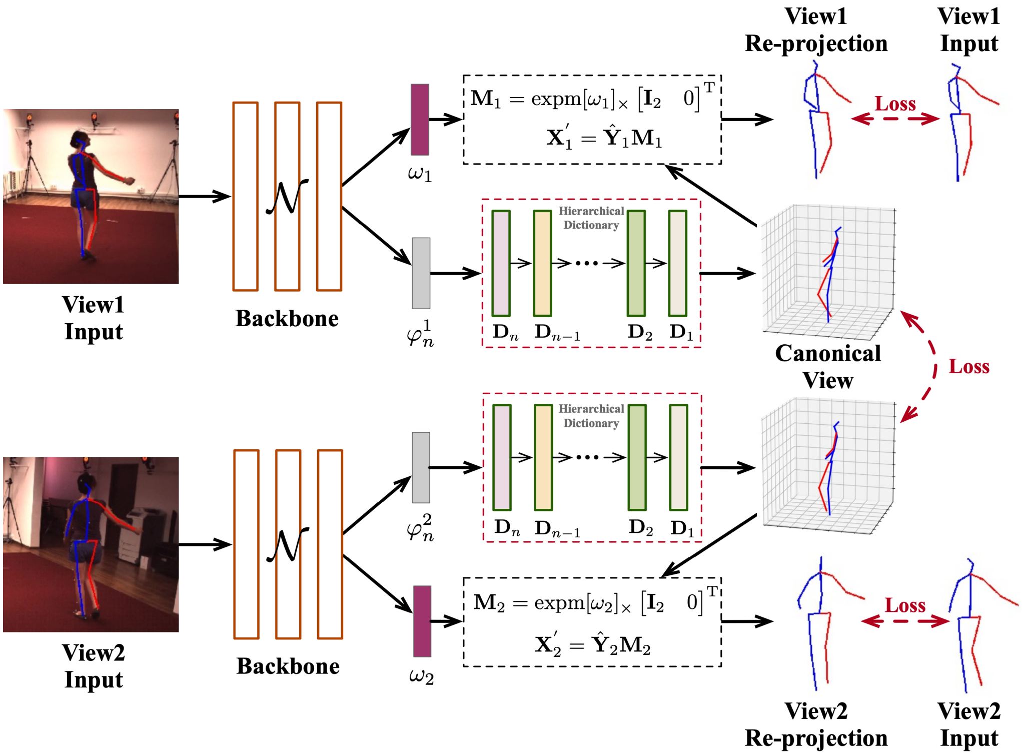

View1
Re-projection
View1
Input
$\mathbf{M}_1 = \text{expm}[\omega_1]_\times \, [\mathbf{I}_2 \quad 0]^{\mathrm{T}}$
$\mathbf{X}_1^{'} = \hat{\mathbf{Y}}_1\mathbf{M}_1$
Loss
$\mathcal{N}$
ω_1
Hierarchical
Dictionary
$\mathbf{D}_n$ $\mathbf{D}_{n-1}$ $\mathbf{D}_2$ $\mathbf{D}_1$
View1
Input
Backbone
φ_n^1
Canonical
View
Loss
Hierarchical
Dictionary
$\mathbf{D}_n$ $\mathbf{D}_{n-1}$ $\mathbf{D}_2$ $\mathbf{D}_1$
$\mathcal{N}$
φ_n^2
$\mathbf{M}_2 = \text{expm}[\omega_2]_\times \, [\mathbf{I}_2 \quad 0]^{\mathrm{T}}$
$\mathbf{X}_2^{'} = \hat{\mathbf{Y}}_2\mathbf{M}_2$
Loss
View2
Input
Backbone
ω_2
View2
Re-projection
View2
Input

Figure 4.1 : Architecture of the consistent factorization network.

ordinate ω. In the following, we will introduce the proposed consistent factorization network in Section 4.4, and introduce how to obtain the hierarchical dictionary by solving an NRSfM minimization problem in section 4.5.

4.4 Consistent Factorization Network

In order to predict the coefficients of 3D human poses and camera viewpoints, we design a deep neural network $\mathcal{N}$ to factorize the two components from input 2D poses. The architecture of the proposed network is shown in Figure 4.1. In detail, the backbone of the network takes the concatenated 2D coordinates of P human joints as inputs. And then, it is followed by four blocks that are surrounded by residual connections. For each block, several fully connected layers (1024 channels) followed by Batch Normalization, Rectified Linear Units, and Dropout, are stacked for efficiently mapping the 2D pose features to high-level features. After that, the high-level features are fed into two specific networks to estimate φ_n and ω. The two networks have the same architecture that is composed of two fully connected layers and a ReLU layer. With φ_n, we feed it to a decoder, which serves as the hierarchical dictionary, to obtain the canonical 3D human pose $\hat{\mathbf{Y}}$. Also, we can calculate the camera rotation matrix $\mathbf{R}$ using ω. Finally, we can obtain the 3D human pose under the camera coordinate $\mathbf{Y} = \hat{\mathbf{Y}}\mathbf{R}$.

4.4.1 Consistent Factorization Contraint

The network will suffer the projection ambiguity problem if training the network simply using the re-projection loss, because it cannot guarantee the human shape and camera viewpoint are fully disentangled. Inspired by the fact that 2D projections from different viewpoints of the same 3D skeleton should have the same canonical reconstruction, we introduce the multi-view information to constrain predicted canonical 3D human poses. Specifically, the network takes as input the 2D skeletons $\mathbf{X}_1, \mathbf{X}_2$ of two randomly selected views captured at the same time. We first measure the re-projection error between input 2D poses and corresponding re-projected 2D poses. Here, we follow the orthogonal projection, and define $\mathbf{M} = \mathbf{R}\begin{bmatrix}\mathbf{I}_2 & 0\end{bmatrix}^{\mathrm{T}}$. Thus, the re-projection loss can be formulated as:

$$\mathcal{L}_{\text{re-proj}} = \|\hat{\mathbf{Y}}_1\mathbf{M}_1 - \mathbf{X}_1\|_2 + \|\hat{\mathbf{Y}}_2\mathbf{M}_2 - \mathbf{X}_2\|_2, \tag{4.3}$$

where $\hat{\mathbf{Y}}_1, \hat{\mathbf{Y}}_2$ and $\mathbf{M}_1, \mathbf{M}_2$ are the predicted canonical reconstructions and camera projection matrixes of two views. Except for the re-projection loss, we design the consistent factorization loss as follows:

$$\mathcal{L}_{\text{cf}} = \|\hat{\mathbf{Y}}_1 - \hat{\mathbf{Y}}_2\|_2, \tag{4.4}$$

which is the l_2 loss between the canonical 3D poses of two different views. This loss forces 2D poses of different views to have the same canonical reconstruction. Meanwhile, the re-projection loss can ensure that the predicted camera rotation matrix

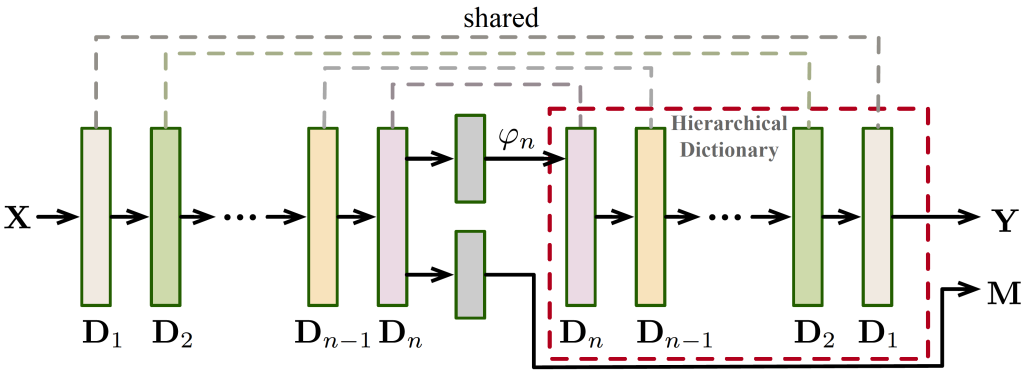


Figure 4.2 : The encoder-decoder network for hierarchical dictionary learning.

projects the 3D poses back to the correct view. In this way, the two components can be fully disentangled. Thus the predicted 3D poses can be constrained from multiple views and overcome projection ambiguity.

4.5 Hierarchical Dictionary Learning

Different from previous methods that learn the dictionary through PCA or sparse coding techniques, we learn the dictionary by solving an NRSfM minimization problem. This is inspired by the most recent work Deep-NRSfM [49], which designs a deep encoder-decoder network to solve the hierarchical dictionary learning problem, where the feed-forward pass through the network can be considered as providing an approximate recovery of multi-layer sparse codes, and the back-propagating through the network can be considered as learning the hierarchical dictionary. In detail, as shown in Figure 4.2, the architecture of the encoder is as follows:

$$\begin{aligned}
\boldsymbol{\Psi}_1 &= \mathrm{ReLU}((\mathbf{D}_1^{\#})^T\mathbf{X} - \mathbf{b}_1 \otimes \mathbf{1}_{3\times 2}),\\
\boldsymbol{\Psi}_2 &= \mathrm{ReLU}((\mathbf{D}_2 \otimes \mathbf{I}_3)^T\boldsymbol{\Psi}_1 - \mathbf{b}_2 \otimes \mathbf{1}_{3\times 2}),\\
&\vdots\\
\boldsymbol{\Psi}_n &= \mathrm{Re\,LU}((\mathbf{D}_n \otimes \mathbf{I}_3)^T\boldsymbol{\Psi}_{n-1} - \mathbf{b}_n \otimes \mathbf{1}_{3\times 2}),
\end{aligned} \tag{4.5}$$

where $\mathbf{D}_1^{\#} \in \mathbb{R}^{P \times 3K_1}$ is the reshape of $\mathbf{D}_1$, and $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are the bias terms. $\boldsymbol{\Psi}_i = \varphi_i \otimes \mathbf{M}$, where $\otimes$ is the Kronecker product. We assume that the camera matrix $\mathbf{M}$ and sparse code φ_n can be extracted from $\boldsymbol{\Psi}_n$ through some function, which is implemented using a linear layer in our work. It is worth mentioning that we predict the exponential coordinates ω to compute the rotation matrix. This is different from Chen and Luecy [49] that adopt SVD to ensure the success of the orthonormal constraint of $\mathbf{M}$, and our implementation is more computationally efficient. With φ_n, we reconstruct the 3D human pose through a decoder:

$$\begin{aligned} \varphi_{n-1} &= \mathrm{ReLU}(\mathbf{D}_n \varphi_n - \mathbf{b}_n'), \\ &\vdots \\ \varphi_1 &= \mathrm{ReLU}(\mathbf{D}_2 \varphi_2 - \mathbf{b}_2'), \\ \tilde{Y} &= \mathbf{D}_1^{\#} \varphi_1. \end{aligned} \tag{4.6}$$

The encoder and decoder are symmetric and share weights. With the encoder-decoder network parameterized by the hierarchical dictionary, we can learn the dictionary by minimizing the re-projection error of all samples in the training set:

$$\min_{\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_n} \sum_i \|\tilde{\mathbf{Y}}_i \tilde{\mathbf{M}}_i - \mathbf{X}_i\|_2, \tag{4.7}$$

where $\tilde{\mathbf{Y}}_i$ and $\tilde{\mathbf{M}}_i$ are the 3D human pose and camera projection matrix of the i-th sample. In this way, we can learn the hierarchical dictionary in an elegant manner.

Although the above encoder network can also predict the 3D pose code and camera viewpoint, it is not robust to the unseen samples.

4.6 Training

During training, we first train the encoder-decoder network to obtain the hierarchical dictionary. After that, the pre-trained decoder, servers as the hierarchical dictionary, is shared by the consistent factorization network, and the network is optimized using both the re-projection loss and consistent factorization loss:

$$\mathcal{L} = \mathcal{L}_{\text{re-proj}} + \lambda \mathcal{L}_{\text{cf}}, \tag{4.8}$$

where λ is a hyper-parameter. In the second step, we train the factorization network, at the same time fine-tuning the hierarchical dictionary with a smaller learning rate to obtain better performance. More training details can be found in Section 4.7.2.

4.7 Experiments

4.7.1 Dataset

We evaluate our method on two popular datasets, which are presented as follows:

Human3.6M [38] is one of the largest datasets for 3D human pose estimation, which is captured by the motion capture system. It includes 3.6 million images captured from 4 calibrated cameras. Moreover, it consists of 15 actions, such as eating, sitting and walking, performed by 11 actors. In our experiments, we use the 17-joint subset following the standard protocol, and we use subjects S1, S5, S6, S7, S8 for training, and S9, S11 for testing.

MPI-INF-3DHP [56] is a recently proposed 3D human pose dataset. Different from Human3.6M, it includes both constrained indoor scenes and complex outdoor scenes. Here, we use the five chest-height cameras and the provided 17 joints for training, and we use the official test set containing 2929 frames (six subjects performing seven actions) for evaluation.

Evaluation Metrics:

For quantitative evaluation, we adopt the common protocol, Mean Per Joint Position Error (MPJPE), which indicates the mean Euclidean distance between the ground-truth and predicted joint positions. Similar to Novotny et al. [60], we restore the scale of the predicted 3D pose before calculating MPJPE Also, we calculate the

Procrustes MPJPE (P-MPJPE), which aligns the estimated 3D pose to the ground truth by a rigid transformation before calculating the MPJPE.

In order to compare with recent methods on the MPI-INF-3DHP dataset, we calculate the Percentage of Correct 3D Keypoints (PCK3D) and Area Under Curve (AUC) [56]. The PCK3D indicates the percentage of joints whose estimated position is within 15cm of the ground truth.

4.7.2 Implementation Details

Data Pre-processing:

The normalization of the input data is crucial for network training. In our work, we normalize the input 2D joint positions through the following steps. First, we set the *Pelvis* joint as the human central joint. Next, we subtract the value of the central joint coordinate from all joints. Finally, we divide the value of all joint coordinates by the scale factor, defined as the mean Euclidean distance of all joints from the central joint.

The Human3.6M dataset only provides four calibrated camera views. In order to augment the dataset, we follow the technique proposed by Fang et al. [23] and simulate a series of virtual camera views. We extend the Human3.6M dataset from four views to twelve views containing eight virtual camera views, and obtain their corresponding 2D poses.

Training Details:

In our work, we train the network in two stages. First, we train the encoder-decoder network on the Human3.6M training data. We set the size of the dictionary at the last level (φ_n) as 10, and the size of the first level (φ_1) as 125. At this stage, we use Adam as the optimizer and train the network for 40 epochs with a learning rate of 0.001. After that, we initialize the hierarchical dictionary of the consistent

Table 4.1 : Per-action P-MPJPE of different variants on the Human3.6M dataset.

	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.
Baseline	123.1	135.9	159.0	129.4	151.3	154.7	114.7	152.7
Baseline+HD	96.8	111.2	105.5	101.4	116.4	129.8	98.7	121.1
Baseline+ADV	54.6	65.3	61.2	69.5	68.0	63.0	83.8	52.4
Baseline+CF	41.9	48.0	47.9	49.0	50.6	64.9	50.6	49.0

	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	**Avg**
Baseline	187.7	210.2	146.5	123.8	146.5	113.4	119.7	144.6
Baseline+HD	144.6	181.3	116.8	109.8	128.2	112.7	110.3	118.9
Baseline+ADV	84.4	84.9	125.8	64.9	67.5	73.5	64.0	72.2
Baseline+CF	60.3	71.7	49.5	54.2	54.8	41.6	47.2	52.1

factorization network with the pre-trained decoder. Meanwhile, the parameters of the consistent factorization network are initialized with Kaiming initializer [33]. For the second stage, we also use Adam as the optimizer, the batch size is set to 1024, and the network is trained for 70 epochs in total. The learning rate of the network backbone starts from 0.001, while the hierarchical dictionary is with smaller value 0.0001 for fine-tuning. The learning rate drops by 0.1 at 50 epochs. The model is implemented using the PyTorch [62] and trained with Intel Xeon E5-2698 2.2GHz and one NVIDIA Tesla V100 GPU.

4.7.3 Ablation Study

Analysis of Consistent Factorization Constraint

In this part, we analyze the effectiveness of the proposed consistent factorization loss. Specifically, we design several variants of our method, and the details of variants are shown as follows:

- **Baseline:** The baseline does not consider the consistent factorization con-

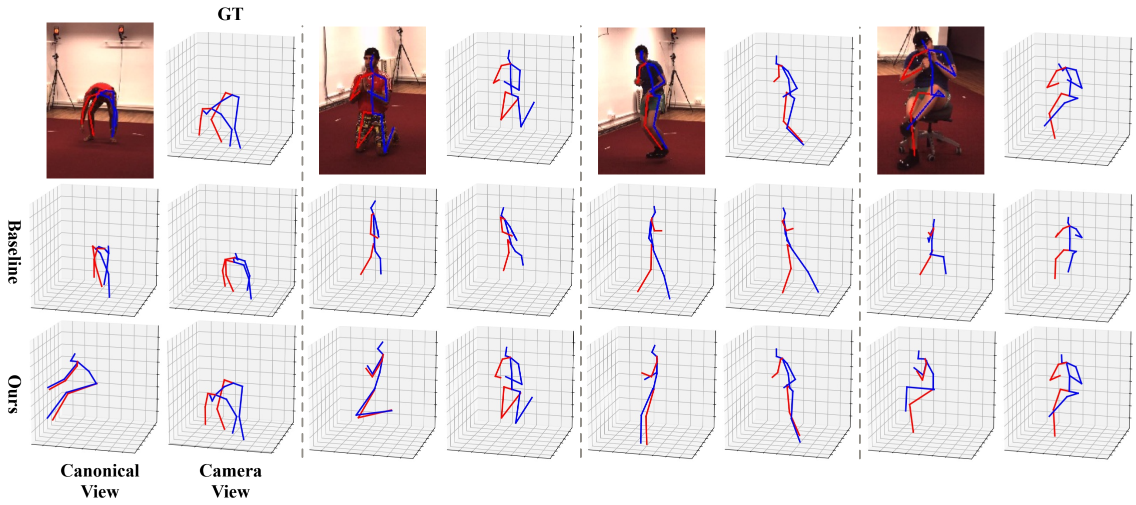


Figure 4.3 : Visualization comparisons of our method versus baseline.

straint and hierarchical dictionary. The baseline is trained simply using the re-projection loss.

- **Baseline+HD:** Different from the baseline, this variant adopts the hierarchical dictionary and pre-trains it using the encoder-decoder network.
- **Baseline+ADV:** Based on the Baseline+HD, this variant adopts adversarial loss to further constrain the 3D pose predictions. It is designed for comparison with the commonly used adversarial loss.
- **Baseline+CF:** This is our proposed method considering both consistent factorization constraint and hierarchical dictionary.

We train all variants on the Human3.6M train set, and Table 4.1 reports per-action P-MPJPE of all variants on the Human3.6M test set. Obviously, our method achieves the best performance among all variants. Compared with the baseline, the pre-trained hierarchical dictionary helps to obtain better results. However, the improvement is limited if only using the hierarchical dictionary. The adversarial loss can further improve performance. In comparison, our method can achieve more

Table 4.2 : Comparisons with recent dictionary-based methods on the Human3.6M dataset.

	MPJPE	P-MPJPE
AIGN [89]	-	97.2
C3DPO [60]	95.6	-
Distill [91]	83.0	57.5
Ours-SD	85.8	57.3
Ours	81.9	52.1

significant improvement and reaches 52.1 (mm) P-MPJPE. This illustrates that consistent factorization constraint is an effective technique to train the 3D human pose estimation network and overcome the projection ambiguity problem. Moreover, our approach does not require any extra un-paired 3D pose annotations.

In addition, we show the visualization comparisons between our method and baseline in Figure 4.3. Here, we visualize the predicted 3D poses of both canonical view and camera view. As we can observe, on the one hand, our approach can generate much more accurate 3D human poses compared with baseline. On the other hand, the canonical reconstructions obtained by our approach maximumly exclude the camera view information. This illustrates that the proposed approach can effectively disentangle the camera view from the 3D human pose, resulting in more accurate results.

Analysis of Hierarchical Dictionary

Here, we analyze the effectiveness of the hierarchical dictionary. Table 4.2 presents the comparisons with recent dictionary-based methods on the Human3.6M

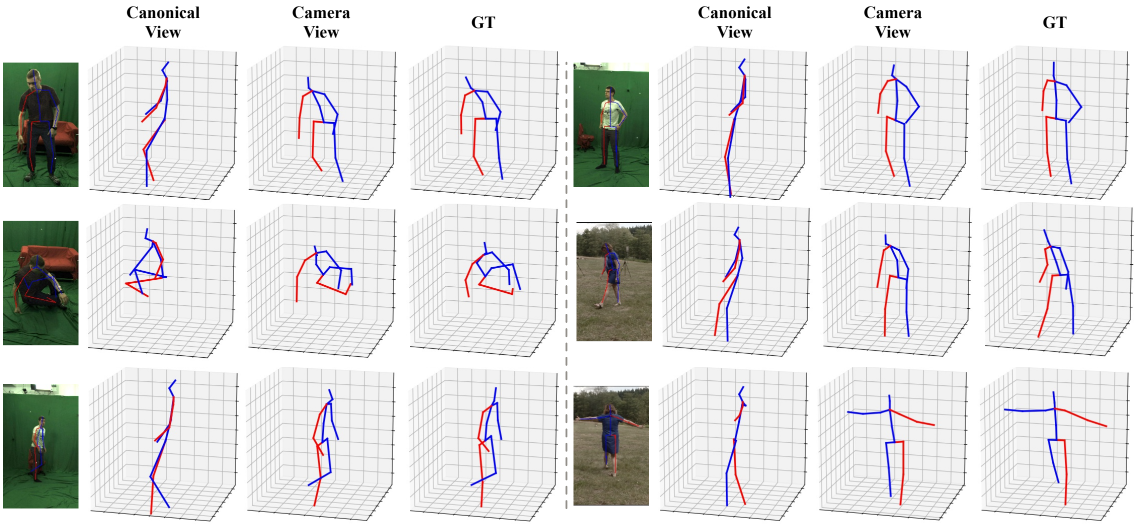


Figure 4.4 : Visualization results of our method (trained on the Human3.6M dataset) on the MPI-INF-3DHP dataset.

dataset. AIGN [89] learns 3D pose dictionary using PCA and adopts adversarial loss as an additional constraint. C3DPO [60] uses a single-level dictionary and learns it with the 3D pose estimation network in an end-to-end manner. Distill [91] is a weakly-supervised method that learns a 3D pose estimation network based on the dictionary learned through NRSfM. As shown in Table 4.2, our method achieves the best result among all.

For further comparisons, we implement a variant (Ours-SD) that replaces the hierarchical dictionary with a single-level dictionary similar to C3DPO. With the consistent factorization constraint, Ours-SD can obtain better performance than C3DPO, 85.8 vs. 95.6 (mm) MPJPE. Moreover, it can be observed that the hierarchical dictionary helps to achieve better results than the single-level dictionary, and MPJPE and P-MPJPE decrease by 3.9 and 5.2 (mm).

Table 4.3 : Comparisons with recent weakly/self-supervised methods on the Human3.6M dataset.

Method	MPJPE	P-MPJPE
Wandt *et al.* CVPR'19	89.9	65.1
Zhou *et al.* ICCV'17	-	64.9
Drover *et al.* ECCV'18	-	64.6
Pavlakos *et al.* CVPR'17	118.4	-
Rhodin *et al.* ECCV'18	-	98.2
Chen *et al.* CVPR'19	-	68.0
Kocabas *et al.* CVPR'19	77.8	70.7
Tung *et al.* ICCV'17	97.2	-
Wang *el al.* ICCV'19	86.4	62.8
Novotny *et al.* ICCV'19	95.6	-
Ours	81.9	52.1
Ours+DA	76.4	47.7

Analysis of Generalization Ability

In order to evaluate the generalization ability of the proposed model, we train the network on the Human3.6M dataset and evaluate it on the MPI-INF-3DHP dataset that includes complex outdoor scenes. We present some visualization results in Figure 4.4, which shows that our method can successfully recover 3D poses on the datasets without being trained on them. Moreover, our method can achieve 70.6% PCK3D and 36.6% AUC in this setting, as shown in Table 4.4.

Table 4.4 : Comparisons with recent weakly/self-supervised methods on the MPI-INF-3DHP dataset.

Method	Trainset	PCK3D	AUC
Zhou *et al.* ICCV'17	H36M	69.2	32.5
Chen *et al.* CVPR'19	H36M	64.3	31.6
Chen *et al.* CVPR'19	MPI	71.1	36.3
Kocabas *et al.* CVPR'19	MPI	71.9	-
Ours	H36M	70.6	36.6
Ours	MPI	74.6	40.4

4.8 Comparisons with State-of-the-art Methods

In this section, we compare our method with recent weakly/self-supervised methods. First, we compare with them on the Human3.6M dataset using MPJPE and P-MPJPE in Table 4.3. Wandt et al. [90], Zhou et al. [111], and Drover et al. [22] are based on adversarial loss and require additional unpaired 3D pose annotations. Tung et al. [89], Wang et al. [91], and Novotny et al. [60] are recent dictionary-based methods. Pavlakos et al. [64], Kocabas et al. [48], Chen et al. [13], and Rhodin et al. [71] require multi-view frames as inputs during training. Here, we report the results that are obtained with and without data augmentation. As shown in Table 4.3, our method obtains state-of-the-art performance, in particular, the P-MPJPE achieves significant improvement compared with previous methods. We found that the gap between MPJPE and P-MPJPE of our method is mostly due to errors of camera viewpoint estimation. It is worth mentioning that the performance can achieve further improvement when using data augmentation, which illustrates that more camera views during training are helpful to improve network performance.

Table 4.4 shows the comparisons with recent methods on the MPI-INF-3DHP dataset. We consider two settings that respectively use Human3.6M and MPI-INF-3DHP datasets as training data, and evaluate the model on the test set following the PCK3D and AUC metrics. As seen, the PCK3D and AUC of our method reach 70.6%/36.6% and 74.6%/40.4% respectively, which outperform previous methods in both settings.

4.9 Conclusion

In this work, we proposed the consistent factorization network for 3D human pose estimation and learned it in a self-supervised manner. The network can fully disentangle the 3D human shape and camera viewpoint through the proposed consistent factorization constraint. It is a simple and effective technique to overcome the projection ambiguity problem, which does not require any extra 3D pose annotations and camera extrinsic parameters. Besides, we introduced the hierarchical dictionary to reconstruct more robust canonical 3D human poses. It was learned through an encoder-decoder network and optimized by an NRSfM minimization problem. After pre-trained, the hierarchical dictionary was further fine-tuned on the consistent factorization network to obtain more accurate 3D pose predictions. Extensive ablation studies on the Human3.6M and MPI-INF-3DHP datasets illustrated the effectiveness and generalization ability of our approach. The experiment results showed that our method can fully disentangle 3D shape and camera viewpoint, and obtained superior performance over recent weakly/self-supervised methods.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Human pose estimation is an important research topic in the computer vision community. This thesis has conducted a study on deep learning based 2D and 3D human pose estimation, and a series of models, from video-based 2D pose estimation to self-supervised 3D pose estimation, have been proposed. The main innovative contributions are summarized as follows:

- We propose the multi-scale TCE module and embed it into the encoder-decoder network architecture for explicitly exploring temporal consistency in videos. The TCE module applies the learnable offset field to capture the geometric transformation between adjacent frames at the feature level. Compared with the existing model-based methods, it can explicitly model the temporal consistency information in an end-to-end network. Compared with the existing post-enhancement methods, it does not require additional optical flow calculations and is more computationally efficient. In addition, we explore the multi-scale geometric transformations at the feature level by integrating the spatial pyramid within the TCE module, which achieves further performance improvements.
- For 3D pose estimation, we propose a self-supervised approach for 3D human pose estimation, which only relies on geometric prior knowledge and does not require any 3D human pose annotations. To this end, we design the

transform re-projection loss, which is an effective technique to exploit multi-view consistency information and constrain the estimated 3D poses during training. Besides, we introduce a root position regression branch to restore the global 3D poses during training. In this way, the network can reserve the scale information of re-projected 2D poses, which can improve the accuracy of the predicted 3D poses. Moreover, this method only relies on geometry knowledge during training, leading to a better generalization ability.

- We propose the consistent factorization network, which fully disentangles the 3D human shape and camera viewpoint to overcome the projection ambiguity problem. To this end, we design a simple and effective loss function using multi-view information to constrain the canonical 3D human pose. Moreover, in order to reconstruct robust canonical 3D human poses, we represent 3D human pose as a combination of a dictionary of 3D pose basis, and adopt geometric information of 3D human poses to learn a hierarchical dictionary from 2D human poses by solving the NRSfM problem. The hierarchical dictionary can be learned without the need for 3D human pose annotations, and has a stronger expression ability compared with the single-level dictionary.

5.2 Future Work

The future research can be conducted in but not limited to the following aspects:

- As for 2D pose estimation, we will attempt to design a unified framework integrating the multi-scale TCE module with the multi-person tracking technique to improve the performance of 2D pose estimation in multi-person videos.

- Besides, the multi-scale TCE module can be extended to the problem of 3D human pose estimation. We will try to extend the multi-scale TCE module to predict three-dimensional offsets and generate temporally enhanced features for predicting 3D human poses.

- As for 3D pose estimation, we will explore the depth map and point cloud data for future work. With the availability of depth cameras and radar sensors on mobile devices, the cost of collecting depth map and point cloud data will become lower. The depth map and point cloud can provide absolute depth information, which can effectively solve the projection ambiguity problem.

Bibliography

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

[2] M. Andriluka, S. Roth, and B. Schiele, “Pictorial structures revisited: People detection and articulated pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1014–1021.

[3] A. Arnab, C. Doersch, and A. Zisserman, “Exploiting temporal context for 3d human pose estimation in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3395–3404.

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[5] 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.

[6] ——, “3d pictorial structures revisited: Multiple human pose estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38,

no. 10, pp. 1929–1942, 2015.

[7] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2017, pp. 468–475.

[8] E. Brau and H. Jiang, "3d human pose estimation via deep learning from 2d annotations," in *Proceedings of the International Conference on 3D Vision*, 2016, pp. 582–591.

[9] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2010.

[10] 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.

[11] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Personalizing human video pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3063–3072.

[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, R. MV, S. Stojanov, and J. M. Rehg, "Unsupervised 3d pose estimation with geometric self-supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5714–5724.

[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets,

atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] L. Chen, H. Ai, R. Chen, Z. Zhuang, and S. Liu, "Cross-view tracking for multi-human 3d pose estimation at over 100 fps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[16] 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.

[17] 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.

[18] A. Cherian, J. Mairal, K. Alahari, and C. Schmid, "Mixing body-part sequences for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2361–2368.

[19] 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.

[20] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2262–2271.

[21] 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.

[22] D. Drover, C.-H. Chen, A. Agrawal, A. Tyagi, and C. Phuoc Huynh, "Can 3d pose be learned from 2d projections alone?" in *Proceedings of the European Conference on Computer Vision Workshop*, 2018, pp. 0–0.

[23] H. Fang, Y. Xu, W. Wang, X. Liu, and S. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, pp. 6821–6828.

[24] 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.

[25] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, no. 1, pp. 67–92, 1973.

[26] K. Fragkiadaki, H. Hu, and J. Shi, "Pose from flow and flow from pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2059–2066.

[27] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, and D. Tran, "Detect-and-track: Efficient pose estimation in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 350–359.

[28] G. Gkioxari, A. Toshev, and N. Jaitly, "Chained predictions using convolutional neural networks," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 728–743.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[30] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 905–10 914.

[31] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge university press, 2003.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[34] ——, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[35] G. Hidalgo, Y. Raaj, H. Idrees, D. Xiang, H. Joo, T. Simon, and Y. Sheikh, “Single-network whole-body pose estimation,” in *Proceedings of the IEEE*

International Conference on Computer Vision, 2019, pp. 6982–6991.

[36] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[37] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2462–2470.

[38] 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.

[39] U. Iqbal, A. Doering, H. Yasin, B. Krüger, A. Weber, and J. Gall, “A dual-source approach for 3d human pose estimation from single images,” *Computer Vision and Image Understanding*, vol. 172, pp. 37–49, 2018.

[40] U. Iqbal, M. Garbade, and J. Gall, “Pose for action - action for pose,” in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition*, 2017, pp. 438–445.

[41] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, “Learnable triangulation of human pose,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7718–7727.

[42] A. Jain, J. Tompson, M. Andriluka, G. W. Taylor, and C. Bregler, “Learning human pose estimation features with convolutional networks,” in *Proceedings of the International Conference on Learning Representations*, 2014.

[43] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, “Towards

understanding action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3192–3199.

[44] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1465–1472.

[45] 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.

[46] 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.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1077–1086.

[49] C. Kong and S. Lucey, "Deep interpretable non-rigid structure from motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1558–1567.

[50] X. Lan and D. P. Huttenlocher, "Beyond trees: Common-factor models for 2d human pose recovery," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 470–477.

[51] C. Li and G. H. Lee, "Generating multiple hypotheses for 3d human pose estimation with mixture density network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9887–9895.

[52] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936–944.

[53] 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 *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[54] Y. Luo, J. S. J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, and L. Lin, "Lstm pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5207–5215.

[55] 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.

[56] 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 *Proceedings of the International Conference on 3D Vision*, 2017, pp. 506–516.

[57] 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.

[58] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 483–499.

[59] B. X. Nie, C. Xiong, and S. Zhu, "Joint action recognition and pose estimation from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1293–1301.

[60] 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 International Conference on Computer Vision*, 2019, pp. 7688–7697.

[61] D. Park and D. Ramanan, "N-best maximal decoders for part models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2627–2634.

[62] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[63] 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.

[64] ——, "Harvesting multiple views for marker-less 3d human pose annotations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6988–6997.

[65] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7753–7762.

[66] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas, “A recurrent encoder-decoder network for sequential face alignment,” in *Proceedings of the European conference on computer vision*, 2016, pp. 38–56.

[67] T. Pfister, J. Charles, and A. Zisserman, “Flowing convnets for human pose estimation in videos,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1913–1921.

[68] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, “Poselet conditioned pictorial structures,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 588–595.

[69] ——, “Strong appearance and expressive spatial models for human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3487–3494.

[70] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, “Cross view fusion for 3d human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4342–4351.

[71] H. Rhodin, M. Salzmann, and P. Fua, “Unsupervised geometry-aware representation for 3d human pose estimation,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 750–767.

[72] 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.

[73] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[74] I. Sárándi, T. Linder, K. O. Arras, and B. Leibe, "How robust is 3d human pose estimation to occlusion?" *arXiv preprint arXiv:1808.09316*, 2018.

[75] S. Sharma, P. T. Varigonda, P. Bindal, A. Sharma, and A. Jain, "Monocular 3d human pose estimation by generation and ordinal ranking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2325–2334.

[76] L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2041–2048.

[77] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 715–731.

[78] J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Thin-slicing network: A deep structured model for pose estimation in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5563–5572.

[79] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[80] 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.

[81] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 529–545.

[82] 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.

[83] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 256–269.

[84] 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.

[85] D. Tome, M. Toso, L. Agapito, and C. Russell, "Rethinking pose in 3d: Multi-stage refinement and recovery for markerless motion capture," in *Proceedings of the International Conference on 3D Vision*, 2018, pp. 474–483.

[86] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 1799–1807.

[87] 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.

[88] H.-Y. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, “Self-supervised learning of motion capture,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5236–5246.

[89] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki, “Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4364–4372.

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

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

[92] C. Wang, H. Qiu, A. L. Yuille, and W. Zeng, “Learning basis representation to refine 3d human pose estimations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, pp. 8925–8932.

[93] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, “3d human pose machines with self-supervised learning,” *arXiv preprint arXiv:1901.03798*, 2019.

[94] Y. Wang and G. Mori, “Multiple tree models for occlusion and spatial constraints in human pose estimation,” in *Proceedings of the European Conference on Computer Vision*, 2008, pp. 710–724.

[95] 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.

[96] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman, “Single image 3d interpreter network,” in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 365–382.

[97] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 466–481.

[98] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, “Learning feature pyramids for human pose estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1281–1290.

[99] W. Yang, W. Ouyang, H. Li, and X. Wang, “End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3073–3082.

[100] 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.

[101] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1385–1392.

[102] ——, “Articulated human detection with flexible mixtures of parts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12,

pp. 2878–2890, 2012.

[103] S. J. Z. Z. C. H. R. Y. D. X. Yang Li, Kan Li, “Geometry-driven self-supervised method for 3d human pose estimation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[104] R. Yeh, Y.-T. Hu, and A. Schwing, “Chirality nets for human pose regression,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2019, pp. 8161–8171.

[105] D. Zhang and M. Shah, “Human pose estimation in videos,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2012–2020.

[106] W. Zhang, M. Zhu, and K. G. Derpanis, “From actemes to action: A strongly-supervised representation for detailed action understanding,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2248–2255.

[107] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, “Semantic graph convolutional networks for 3d human pose regression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3425–3435.

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

[109] F. Zhou and F. De la Torre, “Spatio-temporal matching for human pose estimation in video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1492–1504, 2016.

[110] 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.

[111] 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.

[112] S. Zuffi, J. Romero, C. Schmid, and M. J. Black, "Estimating human pose with flowing puppets," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3312–3319.