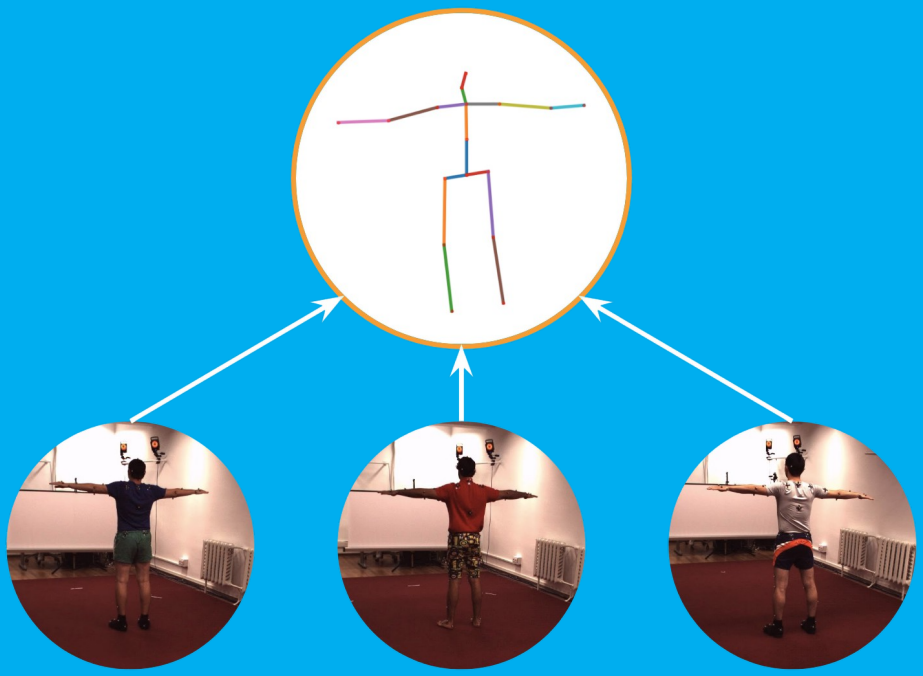
One Pose Fits All

A Novel Kinematic Approach to 3D Human Pose Estimation

Yen-Lin Wu



TU Delft

One Pose Fits All

A Novel Kinematic Approach to 3D Human Pose Estimation

by

Yen-Lin Wu

to obtain the degree of Master of Science
at Delft University of Technology,
to be defended publicly on Monday August 23rd, 2021 at 2:00 PM.

Student number:	4848489	
Project duration:	November 1, 2020 – August 23, 2021	
Thesis committee:	Dr. Jens Kober,	TU Delft, Chair & Supervisor
	Dr. Osama Mazhar,	TU Delft, Daily supervisor
	Dr. Julian Kooij,	TU Delft, External Member

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

3D human pose estimation is a widely researched computer vision task that could be applied in scenarios such as virtual reality and human-robot interaction. With the lack of depth information, 3D estimation from monocular images is an inherently ambiguous problem. On top of that, unrealistic human poses have been overlooked in the majority of papers since joint detection is the only focus.

Our work consists of two parts, an end-to-end 2D-3D lifting pipeline and a novel kinematic human model integrated approach. We start with Pose Estimation using TRansformer (PETR), an approach that does not require temporal information and has the attention mechanism to model the inter-joint relationship from RGB images.

In the approach with human model, we emphasize pose similarity rather than focusing on joint detection. We propose a new metric, called Mean Per Bone Vector Error (MPBVE), that evaluates poses regardless of a human body's gender, weight, or age. We introduce Pose Estimation on Bone Rotation using Transformer (PEBRT), a novel approach that regresses rotation matrices for 16 human bones, assuming labeled 2D poses as input. Our human model encapsulates joint angle and bone length constraints. Existing methods treat these constraints as an additional loss term, which does not guarantee realistic final outputs. Our method does not require temporal information or receptive fields to generate kinematically realistic human poses. We demonstrate that PEBRT is capable of delivering comparable results on Human3.6M to existing methods.

The implementation code is available at https://github.com/wuyenlin/pebrt.

Acknowledgements

This thesis marks the final part of my 2-year master study at TU Delft, and it would not have been possible without the supervision of Dr. Jens Kober and Dr. Osama Mazhar. Your motivating guidance has in many ways shaped my attitude towards research.

I would like to express my gratitude towards Shantanu Shivankar, Varun Kotian, Nikhil Nagendra, and Lokin Prasad for their advice, company, and Indian food.

My thanks extend to Vasileios Sfetsios, who provided warm company in this rather tough and lonely times at the dormitory.

I would not have developed such coding skills without constantly exchanging knowledge with Asier Galicia Martínez.

Many thanks to my parents for their unconditional and unequivocal support.

Immense gratitude to Chia-Jou Chu for her continuous encouragement since my undergraduate study in Japan.

Yen-Lin Wu
Rotterdam, August 2021

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

$MPBVE$	Mean Per Bone Vector Error
$MPJPE$	Mean Per Joint Position Error
PCK	Percentage of Correct Keypoints
$SMPL$	Skinned Multi-Person Linear
SVD	Singular Value Decomposition

Symbols

α	Yaw
β	Pitch
γ	Roll
$\hat{\mathbf{B}}_i$	Normalized bone vector of $i-$ bone
$\mathbf{B}_i^o$	Bone vector of $i-$th bone of a T-pose
$\mathbf{B}_i$	Bone vector of $i-$th bone
$\mathbf{E}$	Extrinsic camera matrix
$\mathbf{J}$	Joint coordinates
$\mathbf{K}$	Intrinsic camera matrix
$\mathbf{R}_i^*$	GT Rotation matrix of $i-$th bone
$\mathbf{R}_i^p$	Rotation matrix of $i-$th bone from projected 2D inputs
$\mathbf{R}_i$	Rotation matrix of $i-$th bone
$\mathbf{w}_{punish}$	Array of punishing weights
B_t	Total number of bones
d_k	Dimension of Key
d_{model}	Dimension of model
J_t	Total number of joints

K	Key
N	Number of Encoder layers
Q	Query
V	Value

1 Introduction

Human pose estimation (HPE) has long captivated the attention of researchers in Computer Vision. It is a study of estimating human body configuration from a single image or video, whose application encompasses action recognition [33, 42], human-robot interaction [62, 86], autonomous vehicle [20], surveillance [23], and avatar generation [29].

Recent works have leveraged the power of Convolutional Neural Network (CNN) to achieve 2D human pose estimation [5, 11, 49] from RGB images. In contrast, 3D human pose estimation is a much more difficult task due to limited dataset available and depth ambiguities. Also, body configuration includes high degree-of-freedom joints, resulting in high-dimensional solution space. Occlusion and truncation can lead to temporal incoherency in pose estimation. Rare and complex poses (e.g. yoga and extreme sports) may be more difficult to infer.

Modern works take 3D pose estimation as a coordinate regression problem, which neglects the human body kinematics and often leads to unrealistic results. In other words., they only regress the coordinates of body joints and does not take account of the structured dependency between keypoints. An example is shown in Figure 1.1. Sun *et al.* [65] define a compositional loss function that encodes local bone relationships. Wandt *et al.* [73] propose Kinematic Chain Space (KCS) matrix that asserts consistent bone lengths throughout the entire image sequence. However, accurate MoCap system and prior knowledge (e.g. bone length) on the human object are required. Dabral *et al.* [15] introduce illegal angle loss and symmetry loss to model joint relationship of human pose. Illegal angle loss is limited to only elbow and knee joints since 2 parent links are required to calculate a normal vector whose dot product with lower arm or calf is positive. Xu *et al.* [79] conduct pose refinement on unreliably estimated joints using motion trajectory of a child joint relative to parent joint. This approach is, again, built upon the assumption of accurate MoCap system and is subject to noisy inputs. On top of that, the above methods of adding additional loss terms may help achieve better accuracy but still does not guarantee realistic final 3D outputs. This prohibits the application in simulation environments and avatar control.

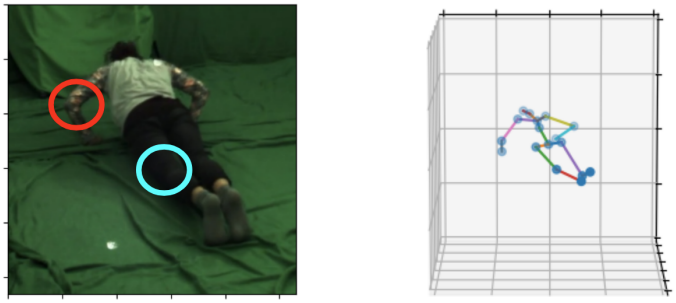


Figure 1.1: An example of kinematically unrealistic output.
Red circle: left lower arm does not have the same length with right counterpart;
cyan circle: knee joint has a unrealistic rotation to the right.

As Zheng *et al.* [83] point out, angle representation is pose-dependent and does not concern body shape. For example, we consider a ROS simulation scenario where controlling a fix-sized Urdf model takes place. Accurate and realistic pose information is a prerequisite to simulation applications. However, pure joint detection struggles to map a single pose to different avatars as every human skeleton has different shape, hence different joint positions. Designing a pipeline that can digest pose information regardless of object's height or shape and guarantees kinematically realistic outputs is a less researched and overlooked topic.

We are motivated to introduce a human kinematic model that is encapsulated with bone length and joint angle constraints. In this work, we incorporate a human kinematic model into deep models that regresses rotation matrix parameters for each bone. We propose "**P**ose **E**stimation via **B**one **R**otation using **T**ransformer (PEBRT)" for monocular 3D human pose estimation. It is a regression approach using a rotation-based representation that incorporates human pose structure. Specifically, our contributions in this work are as follows:

- Our framework does not require receptive field, i.e. multi-frame inputs, to achieve comparable results to existing methods
- Our pipeline recovers rotation matrices from network output and impose them on a kinematic human model
- We propose a new metric that evaluates pose accuracy regardless of human body shape, gender, or age.

To our knowledge, this is the first attempt to formulate human pose estimation with a kinematic model and focus on bone rotations in the entire pipeline. We conduct ablation study to compare the performance on different number of layers of Transformer Encoder. Further experiments show that our model achieve comparable performance to state-of-the-art methods on two widely used 3D human motion datasets.

2

Related Works

This chapter aims to provide a solid ground to the motivation behind our research question by covering 2D and 3D HPE, rotation analysis, and Transformer-based works. Strengths and weaknesses of various 3D HPE methods are also discussed.

2.1. 2D human pose estimation

Accurate 2D estimation is regarded as the prerequisite for accurate 3D prediction. Figure 2.1 is an illustration of commonly used 2D estimators in 3D human pose estimation. Details of each method are given in the next paragraph.

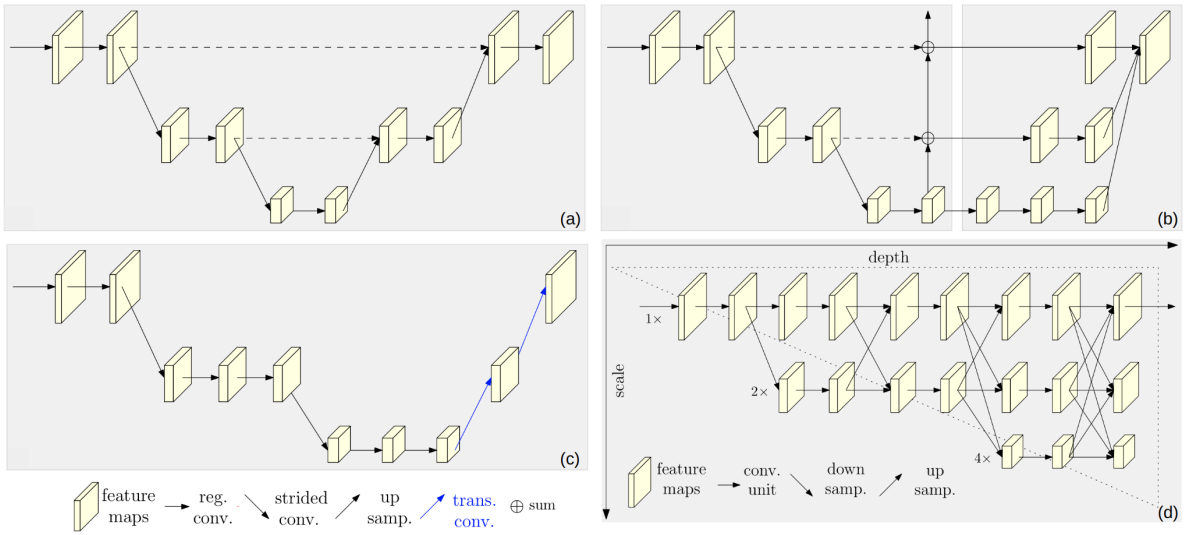


Figure 2.1: Illustration of commonly used 2D estimators [64].
(a) Hourglass [49]; (b) Cascaded pyramid networks [11]; (c) Simple Baseline [77]; (d) HRNet [64].

(a): Newell *et al.* [49] introduce repeated conv-deconv modules named Stacked Hourglass Network (SHN). Through iterative refinement with residual connections in between, spatial information can be preserved and each joint can be localized.

(b): Chen *et al.* [11] propose Cascaded Pyramid Network (CPN) for multi-person pose estimation that consists of two sub-networks, GlobalNet and RefineNet. GlobalNet directly recognizes "easy" keypoints from generated heatmaps, whereas RefineNet explicitly addresses the "hard" keypoints based on an exclusive loss.

(c): Transposed convolution layers were adopted in Simple Baseline [77] to generate high-resolution representations.

(d): In contrast to SHN, Sun *et al.* [64] introduce High-Resolution Net (HRNet) that maintains high-resolution representations throughout the network and aggregates information from parallel sub-networks.

The High-Resolution Network (HRNet) [64] contains 4 convolutional layers and assumes a 256x256 image as input. The output is J_t heatmaps, corresponding to each of the J_t joints. This architecture is used in the pilot implementation of this work as a baseline. It aims to serve as a CNN backbone that extracts 2D keypoints from RGB images.

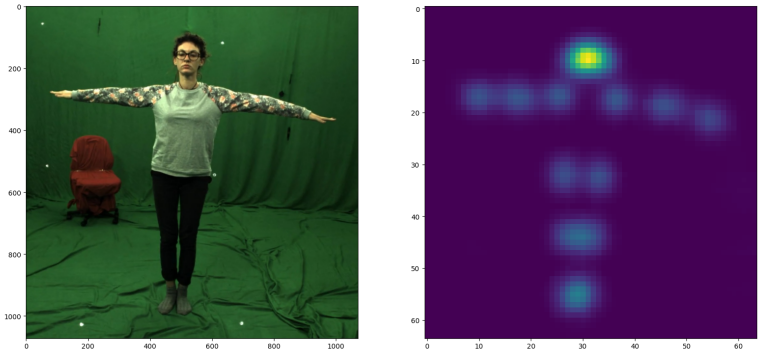


Figure 2.2: An example input and output from HRNet [64].
Left: a RGB image input; right: stacked heatmap corresponding to 17 human joints

2.2. 3D human pose estimation

Some 3D HPE works took advantage of multi-camera settings [19, 27] or depth sensors [56, 62, 86] to achieve accurate estimations. However, such specialized setups are not available to general public, major focus has shifted to developing frameworks for monocular 3D human pose estimation. This section divides deep learning-based 3D HPE methods into 2 categories: model-free and model-based human pose estimation. As mentioned in Chapter 1, predicting depth information from 2D images is an under-constrained task. While training on a model-free architecture is easier, invalid or unrealistic poses can be expected. With kinematic constraints or a reference structure, outputs are guaranteed to have plausible poses. It can, however, take a toll on 3D pose inference time and computing efficiency depending on the optimization approach.

2.2.1. Model-free

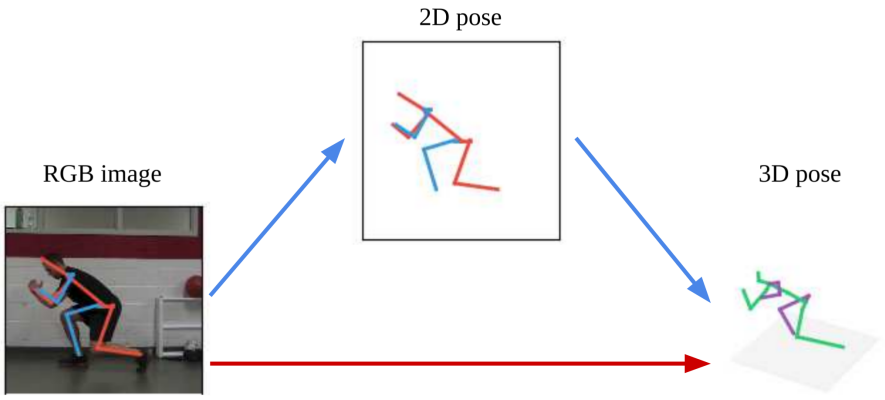


Figure 2.3: Model-free 3D human pose estimation.
Red arrow: Single-stage method; blue arrow: 2D-3D lifting method

Model-free methods can be divided into 2 groups:
1) Direction 3D regression [52, 53, 66], or single-stage methods, skips any intermediate prediction and outputs 3D poses given an RGB image/video as input. The training of such model is easier than 2D-3D lifting while lacking intermediate constraints.

Kanazawa *et al.* [29], Zheng *et al.* [83], and Arnab *et al.* [2] argue that information extracted from images is not leveraged for depth inferences. 2D-3D lifting models can depend solely on input 2D keypoints, thus limiting the final performance. Some of these methods also model 2D-3D correspondences from dataset. As mentioned by Chen *et al.* [9], they incorporate dataset-specific parameters (e.g. camera projection matrix, scale of skeleton, object distance to camera) and achieve good accuracy. However, their performance on in-the-wild image/video would be questionable. Arnab *et al.* [2] reason that overfitting to constrained lab environment is a concern that prevents the model from generalizing well to real-world images.

Pavlakos *et al.* [52] is an example of direction 3D pose estimation. The network predicts per voxel likelihoods for each joint and through repetitive processing and refinement it outputs a final 3D pose. Sun *et al.* [66] argue that heatmap representation is non-differentiable and thus cannot be backpropagated. They propose Integral Pose Regression method to transform heatmaps into joint location coordinate, which is differentiable and in turn allows end-to-end training.

2) 2D-3D lifting [8, 44, 48, 60, 67, 71] is also called two-step/-stage pose estimation. It breaks down the estimation process into: **i.** Producing accurate 2D poses using off-the-shelf 2D pose estimator (e.g. SHN, CPN). **ii.** Lifting the 2D joints to 3D by predicting their respective depth. One of its advantage is that 2D dataset can also be used for training. Chen *et al.* [9] and Wandt *et al.* [73] project 3D predictions back to 2D image space and calculate loss.

2D-3D lifting is an inherently ill-posed problem. While 2D-3D lifting methods sometimes show better performance than the direct regression ones (see Table 2.1, 2.2), their overall accuracy and inference time are dependent and bottlenecked by the 2D estimator. 2D-3D lifting methods can be trained on 2D dataset by projecting 3D poses to 2D image space [47], whereas direct regression methods depend on synthetic data if more training data is required [46].

Chen *et al.* [8] and Rogez *et al.* [60] do not depend on the 2D estimator alone but create a pose library for matching purpose. Chen *et al.* [8] set up a library of 200,000 poses and performed k nearest neighbor search to estimate 3D pose from 2D keypoints. In [60], they pre-process a fixed set of 2D-3D anchor-poses and estimate respective probability to be correct at each location. They perform pose proposal integration (PPI) to aggregate proposals that are close in terms of image location and 3D pose. However, both their performances are limited by the pose library size. Computation speed also depends on the matching algorithm and library size.

Martinez *et al.* [44] is the first one to use deep neural network to realize the concept of lifting 2D to 3D. The lightweight model in [44] set a baseline in the "lifting" category. Veges and Lorincz [71] employ energy optimization based smoothing method to adaptively smoothen 3D poses such that temporarily invisible human does not undermine estimation results. Pavllo *et al.* [55] implement dilated temporal convolutions and receptive fields to capture long-term information.

2.2.2. Model-based

Model-based approaches come in different forms of representation based on the detail and attribute to describe human body shape. Kinematic models output human pose or skeleton, whereas volumetric models provide more information regarding body shape by rendering meshes. Figure 2.4 shows an example of commonly used human models in 3D human pose estimation.

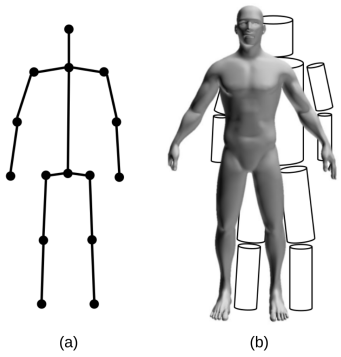


Figure 2.4: Human body models [12]:
(a) Kinematic model; (b) volumetric model

Kinematic model: Kinematic models, also known as skeleton-based models, contain a set of joints locations and their corresponding limb orientations. Constraints on bone length or joint angles can thus be applied to such model. Their simple topology makes them popular among researchers, used in [13, 47, 50, 73]. However, kinematic models fall short of texture or shape information.

Nie *et al.* [50] employ a two-level LSTM architecture to regress depth element for each joint. The first level captures 2D poses from corresponding image patches and human skeletons from 3D pose library, while the second level integrates both global and local features to predict joint depth. Mehta *et al.* [47] is the first to achieve real-time 3D single human pose estimation from a single RGB camera following kinematic skeleton fitting. Their method was claimed to outperform RGB-D cameras (e.g. Kinect [22]) in outdoor scenarios.

Volumetric model: There has been extensive research [2, 3, 40, 54, 69] in human body shape rendering in deep-learning based methods. One of the most widely employed model is Skinned Multi-Person Linear (SMPL) model, introduced by Loper *et al.* [40] and renders a wide range of human body shape using a statistical parametric function, 6890 vertices, and 23 joints.

Pavlakos *et al.* [54] directly predict SMPL parameters given 2D joints and silhouettes. Tripathi *et al.* [69] employ knowledge distillation and trained student network exclusively for SMPL body parameter prediction. Similarly, Arnab *et al.* [2] take advantage of adversarial learning to produce more human models, after which an additional discriminator network distinguishes real models.

2.3. Evaluation metrics

Human3.6M [28] is the most widely used indoor dataset for single person 3D HPE. It contains 3.6 million different human poses collected with 4 digital cameras and consists of 11 professional actors (6 male and 5 female) with different BMI. The actors performed 15 different daily tasks such as walking, smoking, talking on the phone, etc. Provided annotations include 3D joint positions, joint angles, person bounding boxes, and 3D laser scan of each actor Evaluation results are reported in **Mean Per Joint Position Error (MPJPE)**, also known as reconstruction error or 3D error, is the most widely used metric found in literature. It calculates the Euclidean distance from estimated 3D joints to ground truth and average over all joints. It can be written in the form of Equation 2.1.

$$MPJPE = \frac{1}{J_t}\sum_{i=1}^{J} ||\mathbf{J}_i - \mathbf{J}_i^*||_2 \tag{2.1}$$

, where J_t is the total number of joints, $\mathbf{J}_i$ and $\mathbf{J}_i^*$ stand for estimated and ground truth position of joint i. This measurement is reported in millimeters (mm) in 3D or pixel in 2D. MPJPE is considered a generalized baseline metric since it adapts to different dataset that have different number of keypoints. Methods using root-relative pose or absolute pose can be measured using MPJPE. In Human3.6M protocol 1, MPJPE is calculated after aligning the depth of root joint; protocol 2 and 3 is the MPJPE after a rigid transformation using Procrustes Analysis, called P-MPJPE or Reconstruction Error.

MPI-INF-3DHP [45] contains 8 subjects (4 male and 4 female) performing 8 activities, ranging from walking, sitting, sports, etc. There are a total of 1.3 million frames from 14 different angles. They are captured using markerless MoCap system in green screen background, which allows data augmentation such as chroma key compositing.

Percentage of Correct Keypoints (PCK) & Area Under Curve (AUC) are suggested by Mehta *et al.* [45] for more expressive and robust 3D HPE evaluation. AUC is tasked to compute a range of PCK thresholds. Whereas PCK/3DPCK considers a detected joint correct if its distance to ground-truth joint is within a certain threshold, where the default value is $150mm$. Our models are only evaluated on Human3.6M in this work in MPJPE.

Table 2.1: Average 3D Reconstruction error on Human3.6M in Protocol 1 (mm)

Method	Input	Backbone	MPJPE ↓
Model-free			
Chen *et al.* [8] CVPR'17	2D	CPN	82.7
Pavlakos *et al.* [52] CVPR'17	Image	SHN	71.9
Tekin *et al.* [67] ICCV'17	Image	SHN	69.7
Matrinez *et al.* [44] ICCV'17	Image	SHN	62.9
Pavlakos *et al.* [53] CVPR'18	Image	SHN	56.2
Sun *et al.* [66] ECCV'18	Image	ResNet, SHN	49.6
Pavllo *et al.* [55] CVPR'19	2D	CPN	46.8
Model-based			
Mehta *et al.* [47] SIGGRAPH'17	Image	ResNet	80.5
Nie *et al.* [50] ICCV'17	Image	LSTM	79.5
Wandt *et al.* [72] ECCV'18	Image	-	89.9
Cheng *et al.* [13] AAAI'20	Image	HRNet	40.1
Chen *et al.* [10] TCSVT'21	2D	CPN	44.1

Table 2.2: 3D reconstruction error on MPI-INF-3DHP in 3DPCK (%)

Method	Input	Backbone	3DPCK ↑
Model-free			
Matrinez *et al.* [44] ICCV'17	2D	SHN	68.0
Pavlakos *et al.* [53] CVPR'18	Image	SHN	71.9
Pavllo *et al.* [55] CVPR'19	2D	CPN	86.0
Model-based			
Mehta *et al.* [47] SIGGRAPH'17	Image	ResNet	79.4
Wandt *et al.* [72] CVPR'19	Image	-	82.5
Cheng *et al.* [13] AAAI'20	Image	HRNet	84.1
Veges and Lorincz [71] ICONIP'20	2D	-	85.3
Chen *et al.* [10] TCSVT'21	2D	CPN	87.9

2.4. Kinematic constraints

Several works apply previously learnt properties to guarantee valid and realistic 3D poses. Anthropometric priors include bone lengths [13, 59, 73], limb proportions [74], or joint angle constraints [1, 83].

Sun *et al.* [65] address the problem using a regression-based method, instead of pure joint detection. Dabral *et al.* [15] introduce illegal angle loss and symmetry loss to model joint relationship of human pose. Wandt *et al.* [73] propose Kinematic Chain Space (KCS) matrix that implicitly models the kinematic chain of human skeletons without motion priors. The KCS matrix makes it easier to impose constraints on bone length and joint angle. Cheng *et al.* [72] extend the concept in [73] to a temporal application- TKCS, that reports the change in length and angle across different frames in a video. In other words, TKCS ensures the spatial and temporal validity of 3D poses. Zheng *et al.* [83] introduce joint angle prediction constraint in their loss function.

While these methods are able to produce competitive results in accuracy, they assert accurate data input prior knowledge (such as bone length) on the human object. Also, additional terms in loss function does not guarantee realistic final 3D outputs. We design a kinematic human model that contain bone length and joint angle constraints as a novel approach to 3D human pose estimation.

2.5. Deep rotation estimation

There has been prevalent research on rotation estimation, with the output being one of the followings: Euler angles, quaternions, axis-angles, or rotation matrix parameters.

Both Euler angles and quaternions have their limitations in 3D rotation representations, making them difficult for deep neural network to learn [61, 85]. Euler angle representation for 3D rotation shows discontinuity in the case of identity rotation I, i.e. θ can either be 0 or 2π. A common issue with Euler angle (α, β, γ) that causes ambiguity is known as gimbal lock [30]. This occurs when two rotating axes become parallel, one degree of freedom is lost. The order of rotating axis also needs to be specified in advance. Saxena *et al.* [61] point out that a quaternion $\mathbf{q} = (q_x, q_y, q_z, q_w)$ has the antipodal problem, resulting in q and $-q$ to have the same rotation. The axis-angle representation (α, θ) is computationally inefficient in terms of its rotation compositions.

Fisch and Clark [21] propose a 12D over-parameterization called orientation keypoints that model both translation and rotation. However, roll angles are not always available in public dataset, e.g. MPI-INF-3DHP [45]. Zhou *et al.* [85] mathematically prove that such discontinuity and ambiguity stem from 3D rotation representation with 4 or lower dimensions. They further propose general rotation representations by performing Gram-Schmidt orthogonalization. Specifically, they present a continuous $n^2 - n$ dimensional representation for the n dimensional rotation group $SO(n)$. In this work, we utilize this general representation to recover rotation matrices for each bone in the human model.

Another attempt to recover a rotation matrix is by Levinson *et al.* [32], who implement symmetric orthogonalization via Singular Value Decomposition (SVD) on neural network outputs. Cao *et al.* apply such SVD orthogonalization to recover rotation matrices in the study of head pose estimation. Their work also introduce a novel loss function called Mean Absolute Error on Vectors (MAEV). We borrow and modify MAEV to use it as our loss function in our framework.

The above studies in rotation estimation has been widely adopted in 6D object pose estimation and structure from motion (SfM), where the orientation of a single object is estimated. While in human pose estimation, there has not been works estimating bone rotation in the framework. In this paper, we propose a novel approach to estimate the orientation for multiple bones in the human skeleton.

2.6. Transformer-based works

With the introduction of self-attention mechanisms, Liu *et al.* [38] reason that it captures long-range temporal relationships and brings temporal coherency to pose prediction. Since the introduction of Transformer [70], it has stirred up immense interest accompanied with significant progress in language understanding [16, 39, 57] and in image understanding tasks [7, 17, 51, 58, 68].

There has been several attempts to implement Transformer for pose estimation [34, 35, 43, 80, 82]. Zheng *et al.* [82] use a ViT-based architecture to capture spatial and temporal information to lift 2D keypoints to 3D pose. Lin *et al.* [35] combine CNN with Transformer Encoder to output human body meshes. Mao *et al.* [43] propose a regression-based approach which avoids the feature misalignment issue. Li *et al.* [34] employ full encoder-decoder Transformer architecture to perform keypoint/joint regression.

Given its successful implementation on image understanding tasks [7, 17, 36, 75], we are curious about Transformer's potential in handling sequences such as joint coordinates. Though Transformer has demonstrated improved training speed, its inference time struggles to keep up due to its auto-regressive schema in the decoder [81]. Xu *et al.* [78] acquire faster inference time using Transformer by discarding decoder layers. In this work, we use only the encoder part of Transformer with its potential real-time application in mind.

3

Joint Detection Approach

Following the majority of research in human pose estimation, we propose an end-to-end 2D-3D lifting approach for 3D human pose estimation. We employ HRNet as the feature extractor, followed by Transformer Encoders and a fully-connected (FC) layer. This section introduces the cutting-edge model Transformer, camera projection needed to process the dataset, and the architecture of our pilot implementation on human pose estimation.

3.1. What is Transformer?

Sequence-to-sequence (Seq2Seq) problems, such as language modeling and neural machine translation, used to be tackled with Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM). However, their applications are not without limitations. They struggle with long sequence input since the reference window limits how far they can look back in the input. Physical memory constraints also act as a setback against long sequence input. This results in vanishing gradient as important information is *forgotten* after layers and layers of training.

This section introduces the cutting-edge technique proposed by Vaswani *et al.* [70], known for its robust performance in Natural Language Processing (NLP). The working principle of Transformer and its application in Computer Vision, especially in HPE, will be discussed by introducing state-of-the-art research. [70] address the input sequence limit by introducing a model that depends entirely on attention mechanism, dispensing convolution and recurrence. Transformer follows this overall architecture (Figure 3.1) using multi-head attention and point-wise, fully connected layers for both encoder and decoder.

3.1.1. Embeddings

Before feeding a word sequence into Transformer, words need to be converted to vectors for the computer to understand. Word embeddings are regarded as dense representations where words with similar meanings are close to each other and have similar vector representations.

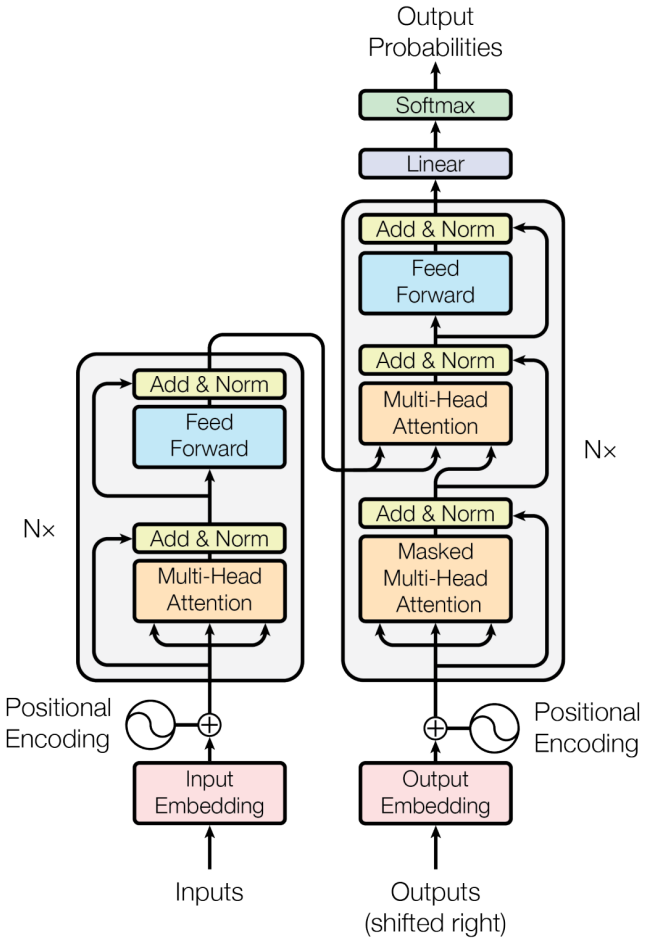


Figure 3.1: Transformer model architecture [70]

3.1.2. Positional encoding

Due to absence of recurrence and convolution, information of relative or absolute position of the token needs to be embedded for Transformer. Without positional information, "Kevin hurt the dog" would have the same representation as "the dog hurt Kevin". [70] use simple sin and cos function to generate positional encoding, as shown in Equation (3.1),(3.2), where pos and i stand for the position and dimension, and d_{model} is the model dimension. The amplitude of the encoding is bounded by -1 and 1 thanks to the trigonometric functions.

Figure 3.2 is a visualization of the positional encoding used in [70]. We inherit the original implementation in this work. The encoded value are added to the vector representation, allowing us to distinguish the order.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{3.1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{3.2}$$

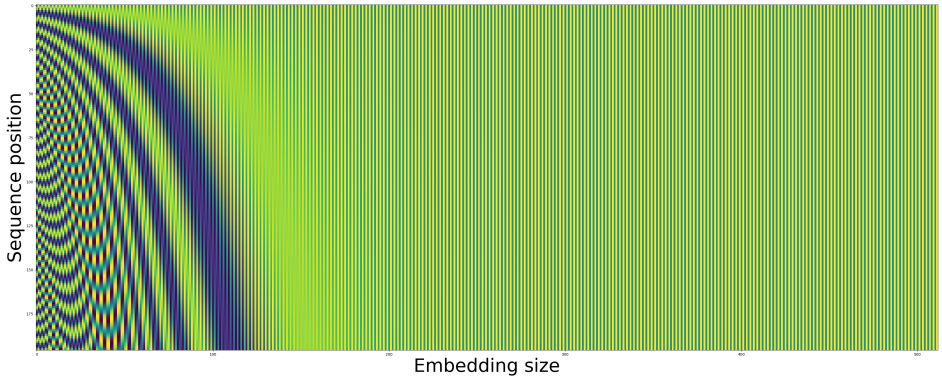


Figure 3.2: Visualization of positional encoding

3.1.3. Multi-head attention

Transformer adopts scaled dot-product attention, as shown in Equation (3.3). The output is a weighted sum of values (V), whose weight is obtained by the dot-product of the query (Q) with all the keys (K).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.3}$$

where Q, K, V are matrices for queries, keys, and values, d_k is the dimension of K, and $\frac{1}{\sqrt{d_k}}$ serves as a scaling factor.

For each single word in the sequence, self-attention generates an attention vector that models its contextual relationship with all other words in the same sequence. This not only solves the long-range dependency issues that RNN faced with, but also works under parallelization, making it more computationally efficient [1] than convolution and recursive operations. This brings us to the definition of multi-head attention, where attention layers are stacked in parallel.

3.1.4. Encoder

Each encoder block includes a multi-head attention and a feed-forward layer. The multi-head attention layers outputs multiple attention vectors. The feed-forward layer is tasked to convert them into a single attention vector by normalization for the succeeding encoder or decoder block to process. A total number of 6 layers of encoder was used in [70]. We conduct ablation studies on layer numbers of encoder in this work.

3.1.5. Decoder

The masked multi-head attention generates attention vectors for each word in the output sequence. Instead of using every element like encoder does, the masked multi-attention can only refer to previous outputs. Any future elements are masked by setting their values to zero. In the next attention layer, we have output from the last encoder transformed into

[1] Table 1 of [70] gives computing complexity of convolutional, recurrent, and self-attention model.

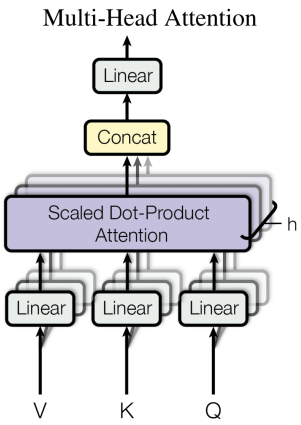


Figure 3.3: Multi-head attention [70] stacks h layers of scaled dot-product attention in parallel before concatenating their attention scores

matrix K and V, with Q being the output from masked multi-head attention. This is the process where the relationship between input and output sequence are mapped.

Similar to encoder, multiple attention vectors are normalized to a single one. [70] use 6 layers of decoder before output to the final linear layer.

3.2. Architecture of PETR

Our first model follows an end-to-end approach- inferring 3D keypoints directly from a monocular image. The HRNet takes in an RGB image and outputs 17 heatmaps, each highlights a different joint location. By unraveling the index, the 2D coordinates of each joint can be obtained, giving us an array of (17,2). Transformer encoder assumes concatenated (x,y) coordinates of the 17 joints that are added with positional encoding. The output of encoder gives us the same shape of array, (1,34). Finally, the fully-connected (FC) layer and $tanh$ activation function outputs a prediction of the 3D pose. Figure 3.4 shows an instantiation of our architecture.

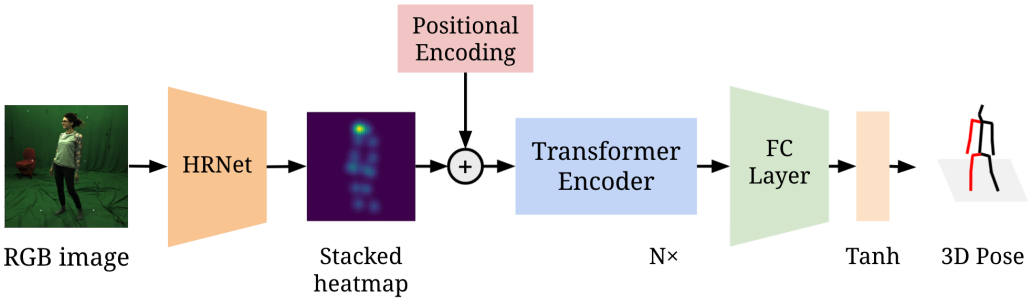


Figure 3.4: Architecture of PETR.
The pipeline accepts RGB image as input, generates a heatmap of human joints, feed them into N layers of Transformer Encoders followed by a linear layer and a $tanh$ activation function, and outputs a single 3D pose.

Xu *et al.* [78] investigate the effect on the number of encoder and decoder layer in word translation tasks. Their experiments show that trading decoder for encoder layers accompanies marginal loss in accuracy, less number of parameters, and more than 3 times faster inference time. We only make use of the encoder part of Transformer due to the concern of inference speed.

3.3. Dataset preprocessing

We standardize the dataset before using since they 1) come in different joints order, and 2) are captured in world coordinates. The output from pre-trained HRNet is in COCO format; whereas Human3.6M [28] and MPI-INF-3DHP [45] comes in different order. We rearrange and follow the joints order shown in Figure 3.5.

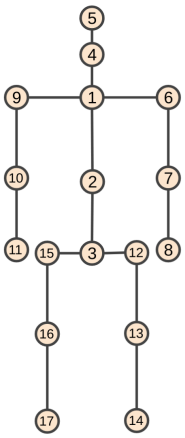


Figure 3.5: Joints order
(1. manubrium, i.e. midpoint of left and right clavicles; 2. mid-spine; 3. root; 4. neck; 5. face; 6. left shoulder; 7. left elbow; 8. left wrist; 9. right shoulder; 10. right elbow; 11. right wrist; 12. left hip; 13. left knee; 14. left ankle; 15. right hip; 16. right knee; 17. right ankle)

Since intrinsic matrix is available in the majority of commercialized cameras, we process 3D keypoints in camera coordinates, instead of in world coordinates. The advantage is that the neural network will not predict the extrinsic matrix, which is unknown without inference, and focus only on joint detection.

The 3D annotations in Human3.6M and MPI-INF-3DHP are given in world coordinates. For the Human3.6M dataset, we borrow the processed file used in VideoPose3D [55]. It includes 2D keypoints in pixel coordinates and 3D keypoints in camera coordinates, where the root joint is centered at origin $(0,0,0)$.

On the other hand, 3D keypoints in MPI-INF-3DHP have to be manually projected. Since intrinsic matrix are given, 3D keypoints in camera coordinates can be calculated. The equation to relate 2D and 3D coordinates is given below:

$$\mathbf{x} = \mathbf{P}\mathbf{X} \tag{3.4}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.5}$$

, where $\mathbf{x}$ is a pixel coordinates (u, v) and $\mathbf{X}$ is a world coordinates (X, Y, Z) in their homogeneous form, and $\mathbf{P}$ is a 3×4 camera matrix. Camera matrix $\mathbf{P}$ is commonly known as the product of the intrinsic matrix $\mathbf{K}$ and extrinsic matrix $\mathbf{E}$.

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} f & s & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array}\right] \\ &= \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \\ &= \mathbf{K}\mathbf{E} \end{aligned} \tag{3.6}$$

, where f is the camera focal length, (p_x, p_y) is the principle point, s is the skew coefficient, and $\mathbf{R}$, $\mathbf{t}$ constitutes the extrinsic matrix , or rigid transformation, between world and camera coordinate systems. Since it was found in [55] that the lens distortion has negligible impact on the pose estimation metric, we assume 0 lens distortion in our work.

With the basic equations of camera projection and the method proposed by Lepetit *et al.* [31], 3D keypoints in camera coordinates $\mathbf{X}_c$ can be recovered as follows.

$$\mathbf{X}_c = \mathbf{E}\mathbf{X} = \mathbf{K}^{-1}\mathbf{x} \tag{3.7}$$

4

Kinematic Model Approach

This section addresses the architecture of Pose Estimation on Bone Rotation using Transformer (PEBRT). Instead of accurately detecting body joints from an image, we postulate that pose estimation regardless of input skeleton size be the end goal. Hence in this section, PEBRT is proposed to predict the rotation matrix parameters for each of the 16 bones in a human body. The architecture inherits from PETR- Transformer encoders and a fully-connected layer.

4.1. Architecture of PEBRT

With recent advent 2D joint detection methods such as Mask R-CNN [26] and CPN [11], more research turns to 3D keypoints inference given 2D inputs. This model follows the classic 2D-3D lifting approach that predicts root-relative 3D coordinates of joints associated to human skeletons. We use N layers of the original Transformer encoder, followed by a linear layer and $Tanh$ activation function that outputs $B_t * 6$ elements. This stands for 6 relevant elements of a rotation matrix of B_t bones, to which the Gram-Schmidt process [85] is applied to recover the rotation matrix for each bone. B_t is 16 in our case.

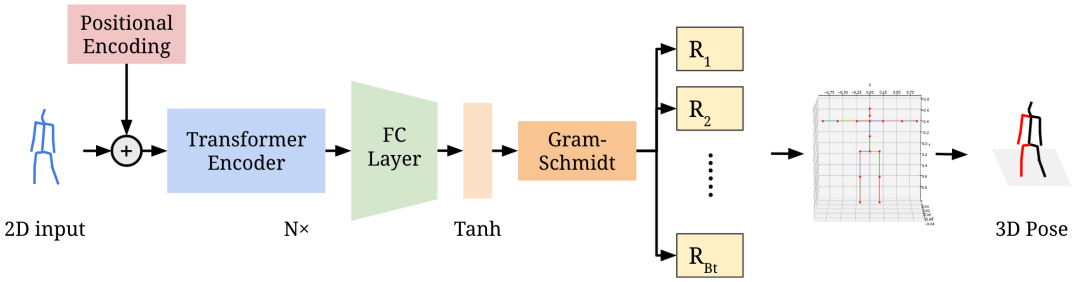


Figure 4.1: Architecture of PEBRT.
The pipeline takes in a single 2D pose as input, feeds it through N layers of Transformer Encoders followed by a linear layer. Gram-Schmidt process is applied to the linear layer output to recover rotation matrices for each bone. The rotation matrices are applied to our human model for the final 3D pose.

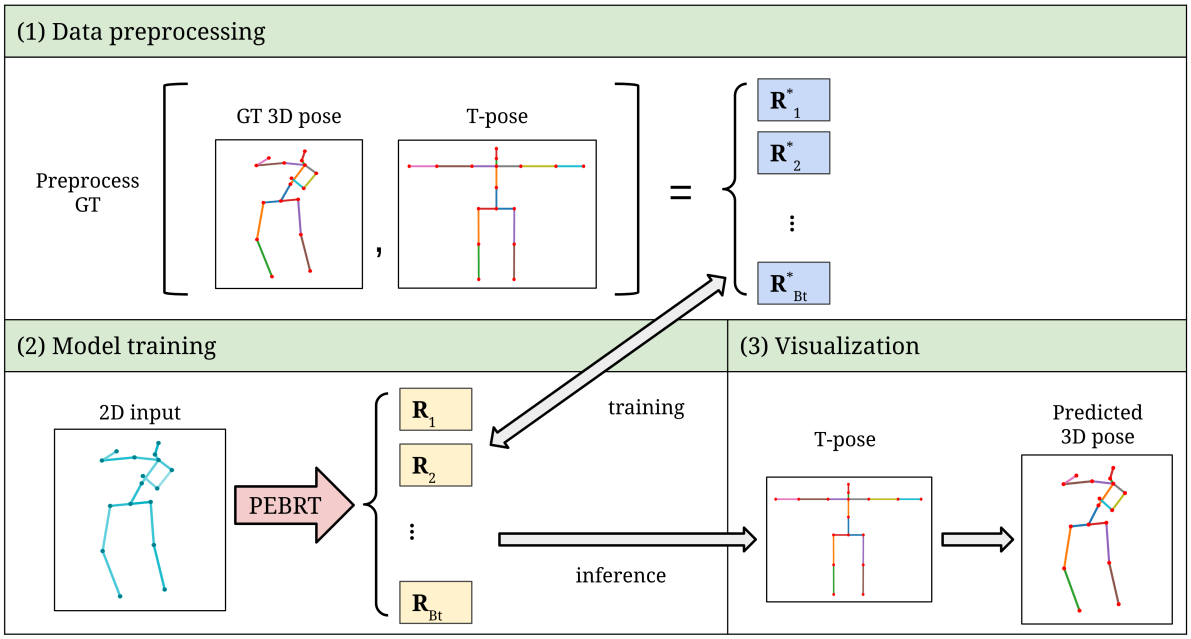


Figure 4.2: An overview of PEBRT pipeline.
(1) Preprocess data to obtain GT rotation matrices for each bone. (2) The network outputs rotation matrices during training. (3) Predicted rotation matrices can directly be applied to human model to make the pose.

An overview of PEBRT is shown in Figure 4.2. During data preprocessing in (1), a given GT 3D pose is used to obtain rotation matrices for each bone taking T-pose as reference. Gram-Schmidt process is applied to the network output, where the GT rotation matrices $\mathbf{R}_i^*$ are used to calculate MAEV loss [6] during training (2), where i stands for $i-$th bone. Finally in (3), the predicted rotation matrices $\mathbf{R}_i$ are applied to a T-pose human model that rotates each bone accordingly during inference. Details of each step are given in the following sections.

4.2. Human Kinematic model

To create our novel kinematic model, we define the body segment lengths as given in [18] and the joint range of motion as given in Appendix B of [25]. We follow the convention of rotation whose yaw, pitch, roll are denoted by α, β, γ, respectively standing for rotation about $z-$, $y-$, and $x-$axis. The human model is postured in T-pose by default (see Figure 4.3 (left)). We further define the bone vectors in the order shown in Figure 4.3 (right). The information of parent/child bone to decide the punishing weight on each bone (see §4.4).

Table 4.1 gives the following information that is present in the model:

1. body segment lengths in a fraction of body height
2. joint constraints (in degrees) that are present in the model
3. whether each bone is categorized as a parent (P) bone or a child (C) bone

Note that the joint angles of a child bone is treated as relative rotation to its parent bone.

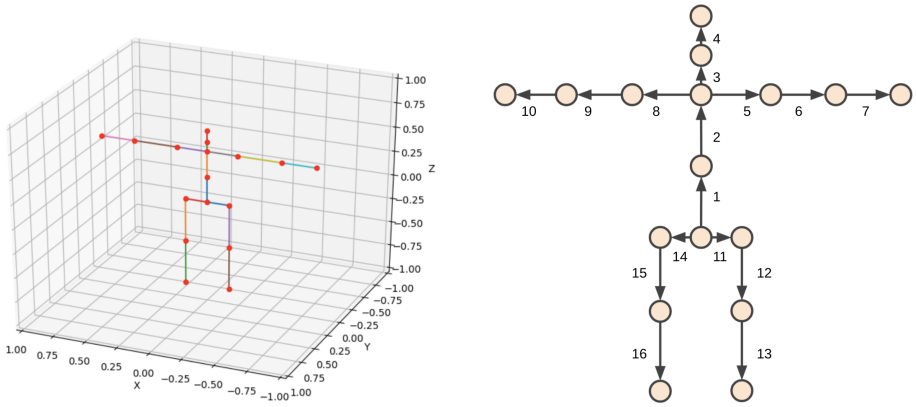


Figure 4.3: Human kinematic model in 3D view (left) and the order of bone vectors (right)
(1. lower spine; 2. upper spine; 3. neck; 4. head; 5. left clavicle; 6. left upper arm; 7. left lower arm; 8. right clavicle; 9. right upper arm; 10. right lower arm; 11. left pelvis; 12. left thigh; 13. left calf; 14. right pelvis; 15. right thigh; 16. right calf)

Table 4.1: Body segment lengths in a fraction of body height H, joint constraints in Euler angles (deg), and categories of parent (P) or child (C) bone

Body Part	Length	$[\alpha_{min}, \alpha_{max}]$	$[\beta_{min}, \beta_{max}]$	$[\gamma_{min}, \gamma_{max}]$	P/C
Head	0.130H	[-70,70]	[-35, 35]	[-55, 80]	C
Neck	0.052H	[0, 0]	[0, 0]	[0, 70]	P
Upper spine	0.144H	[0, 0]	[0, 0]	[0, 95]	C
Lower spine	0.144H	[-30, 35]	[-35, 35]	[-30, 75]	P
Upper arm*2	0.186H	[-45, 130]	[-90, 130]	[-90, 180]	P
Lower arm*2	0.146H	[0, 150]	[0, 0]	[0, 0]	C
Thigh*2	0.245H	[-45, 45]	[-20, 50]	[-30, 120]	P
Calf*2	0.246H	[0, 0]	[0, 0]	[0, 160]	C

4.3. Rotation estimation

4.3.1. Recovering rotation matrix

Zhou *et al.* [85] suggest that functions of stronger continuity properties lead to lower approximation error. In other words, discontinuity hinders the performance of neural networks. They prove that 3D rotation representation is discontinuous in 4 or lower dimension, which indicates that it is inappropriate to use Euler angles or quaternion. Based on the same argument, a 3D rotation matrix has 9 elements and thus do not have such discontinuity issue.

A 3D rotation matrix is characterized as an orthogonal matrix with a determinant of +1. All rotation matrices form a group called special orthogonal group $SO(n)$, where n is the dimension. Levinson *et al.* [32] assume 9D network output, forming a noisy predicted matrix $\mathbf{R}$, and utilize SVD to find the the closest rotation matrix $\hat{\mathbf{R}}$. However, $det(\hat{\mathbf{R}})$ is not guaranteed to be +1 depending on how noisy the original matrix is. [85] show a general form of 3D rotation representation using Gram-Schmidt process with 6D overparameterization. The mapping f_{GS} from 6D representation to $SO(3)$ can be obtained using Equation (4.1).

$$f_{GS}\left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix} \tag{4.1}$$

where

$$b_i = \begin{bmatrix} N(a_1) & if & i=1 \\ N(a_2-(b_1\cdot a_2)b_1) & if & i=2 \\ b_1\times b_2 & if & i=3 \end{bmatrix}^T$$

The 6D representation is beneficial to neural network since the Gram-Schmidt process ensures the matrix orthogonality. It is a better approach than directly predicting 3x3 rotation matrix since orthogonalization has to be done as a post-process, which has been reported to have higher error [85] and also restricts applications in forward kinematics.

4.3.2. Obtaining ground-truth rotation matrix

In order to extract useful bone information, we use each ground-truth (GT) 3D pose and a human model in T-pose to obtain the rotation matrix of each bone. In other words, we calculate the rotation matrix $\mathbf{R}_i^*$ that rotates i-th bone vector of T-pose $\mathbf{B}_i^o$ onto GT i-th bone vector $\mathbf{B}_i^*$, i.e. $\mathbf{B}_i^* = \mathbf{R}_i^*\mathbf{B}_i^o$. This idea is visualized in Figure 4.4.

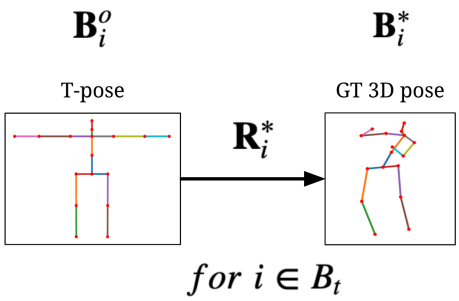


Figure 4.4: Rotation matrices $\mathbf{R}_i^*$ are obtained such that $\mathbf{B}_i^* = \mathbf{R}_i^*\mathbf{B}_i^o$ for $i \in B_t$, where $\mathbf{B}_i^*$ and $\mathbf{B}_i^o$ stand for the $i-$th bone of GT pose and T-pose, and B_t is total number of bones.

Essentially, we are calculating the rotation matrix that relates the two given vectors. Let us first denote $\mathbf{v} = (v_1, v_2, v_3) = \mathbf{B}_i^o \times \mathbf{B}_i^*$ and $c = \mathbf{B}_i^o \cdot \mathbf{B}_i^*$. The rotation matrix $\mathbf{R}$ is given by

$$\mathbf{R} = \mathbf{I} + [\mathbf{v}]_\times + [\mathbf{v}]_\times \frac{1-c}{s^2} \tag{4.2}$$

$$[\mathbf{v}]_\times := \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{4.3}$$

, where $[\mathbf{v}]_\times$ is the skew-symmetric matrix of $\mathbf{v}$, and $s = ||\mathbf{v}||$ is the norm of v. This process is done for each bone in each 3D pose in the dataset.

4.3.3. Verifying implementation

To verify whether the method in §4.3.2 is functional, we load an image with its corresponding GT 3D keypoints, with which GT rotation matrices can be inferred. The GT rotation matrices are applied to our human model with height h set to $1.5m$ and $1.9m$.

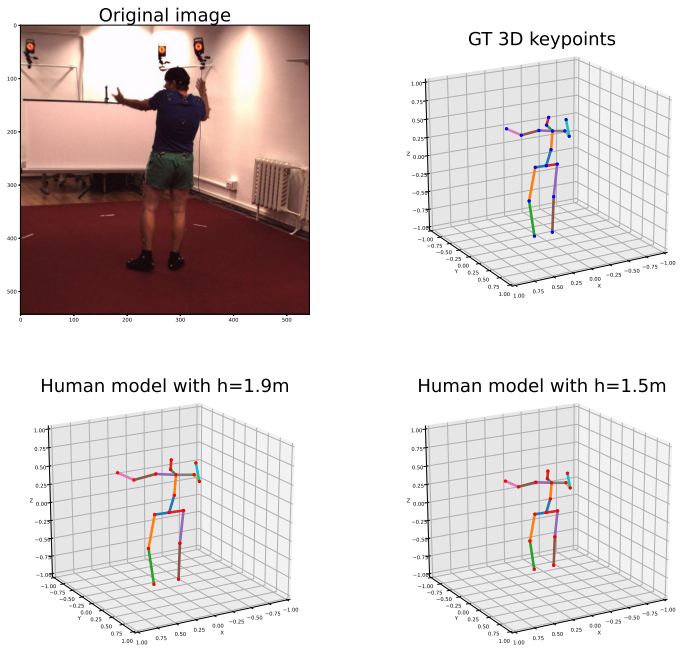


Figure 4.5: An example of imposing GT information on our human model.
(Top left) Original image; (top right) GT 3D keypoints corresponding to the image;
(bottom left & right) Inferred GT rotation matrices applied to human model with $h = 1.9m$ and $h = 1.5m$.

4.4. Loss function

Cao *et al.* [6] introduce Mean Absolute Error of Vector (MAEV) and use it as their evaluation metric in head pose estimation. We modify this as our loss function by introducing punishing weights on each bone depending on whether their rotation exceeds the pre-defined limits in Table 4.1.

To start with, we have 6D network output $[\mathbf{a}_1, \mathbf{a}_2]$ that are mapped to 3D rotation matrix using Gram-Schmidt process, as explained in § 4.3.1. We see a rotation matrix $\mathbf{R}$ as a set of 3 orthogonal column vectors and rewrite Equation (4.1) as follows.

$$f_{GS}([\mathbf{a}_1, \mathbf{a}_2]) = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] = \mathbf{R} \tag{4.4}$$

Figure 4.6 is an illustration of how our modified MAEV works. The discrepancy between column vectors of recovered matrix $\mathbf{R}$ and GT matrix $\mathbf{R}^*$ are denoted as d_1, d_2, and d_3. Their Frobenius norm is hence equivalent to $\sqrt{d_1^2 + d_2^2 + d_3^2}$.

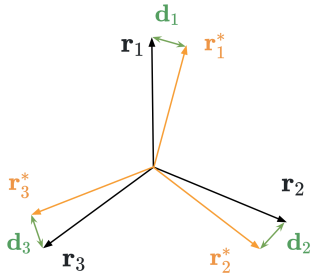


Figure 4.6: MAEV, a novel loss function proposed by Cao *et al.* [6]. Given a recovered matrix $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ and its ground-truth counterpart $\mathbf{R}^* = [\mathbf{r}_1^*, \mathbf{r}_2^*, \mathbf{r}_3^*]$, the Frobenius norm of the difference between $\mathbf{R}$ and $\mathbf{R}^*$ is $\sqrt{d_1^2 + d_2^2 + d_3^2}$

At the meantime, recovered matrix $\mathbf{R}_i$ is decomposed into Euler angles using the technique in [63]. Algorithm 1 accepts $\mathbb{R}_{B_t} = \{\mathbf{R}_i | i \in 1, 2, ..., B_t\}$ as input and outputs $\mathbf{w}_{punish} = \{w_i | i \in 1, 2, ..., B_t\}$, with w_i corresponding to a total number of bones B_t.

$$\mathscr{L} = \frac{1}{B_t} \sum_{i=1}^{B_t} w_{(i)} * ||\mathbf{R}_i - \mathbf{R}_i^*||_2 \tag{4.5}$$

Algorithm 1: Calculating punishing weight for each bone

Data: $\mathbb{R}_{B_t}$; // rotation matrix for each bone
Result: $\mathbf{w}_{punish}$; // punishing weight for each bone
for *i-th bone* **do**
  $a, b, r \leftarrow Decompose(\mathbf{R}_i)$;
  if *child bone* **then**
    /* get relative rotation angles w.r.t parent bone */
    $\hat{a}, \hat{b}, \hat{r} \leftarrow relative(a, b, r)$;
  else
    /* use absolute rotation angles */
    $\hat{a}, \hat{b}, \hat{r} \leftarrow a, b, r$;
  if *($\hat{a}$, $\hat{b}$, $\hat{r}$) not within constraints* **then**
    $w_i \leftarrow 2.0$;
  $w_i \leftarrow 1.0$;

4.5. Novel evaluation metric

Since our framework is based on a kinematic model with fixed bone length, conventional metric on joint position error would not be an appropriate evaluation approach. We propose Mean Per Bone Vector Error (MPBVE) to assess human pose accuracy, regardless of human body shape, age, or gender.

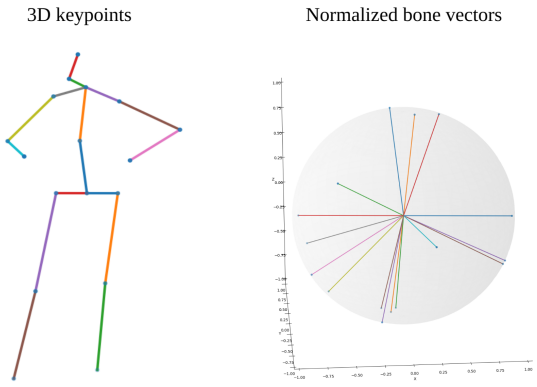


Figure 4.7: Preprocessing 3D keypoints.
(1) Obtain bone vectors from joint coordinates; (2) Normalize each vector to $1m$.

To implement this metric on given 3D keypoints, the following procedures are applied:

- Vectorize - obtain bone position vectors based on a predefined order
- Normalize - normalize each vector to $1m$

This metric accepts ordinary 3D estimation outputs in the shape of $(J_t, 3)$, where J_t is total number of joints. Bone position vectors can be calculated and normalized per human pose, resulting in the shape of $(B_t, 3)$, where B_t is total number of bones. Same operation applies to the ground-truth data. Hence, the $L2$ norm of the predicted and ground-truth bone vectors can be calculated. This generic metric (see Equation (4.6)) works for both 2D and 3D setup under a pre-defined bone order. In this work, we use the bone order defined in Figure 4.3 (right).

$$MPBVE = \frac{1}{B_t} \sum_{i=1}^{B_t} ||\hat{\mathbf{B}}_i - \hat{\mathbf{B}}_i^*||_2 \tag{4.6}$$

, where $\mathbf{B}_i$ is the i-th bone vector and $\hat{\mathbf{B}}_i = \frac{\mathbf{B}_i}{||\mathbf{B}_i||}$ is normalized to $1m$, $\hat{\mathbf{B}}_i^*$ is the normalized GT bone vector, and B_t is the total number of bones.

4.6. Additional loss term

Instead of using only MAEV as loss function, we propose 2 different approaches to improve the model performance.

Including joint position error When using only MAEV as loss function, we faced an issue that is mismatched camera angle, as shown in Figure 4.8. We propose a solution by including a weighted joint position error (MPJPE) to the loss function. This would ideally match the human model to its correct orientation yet not enough to deteriorate the overall accuracy.

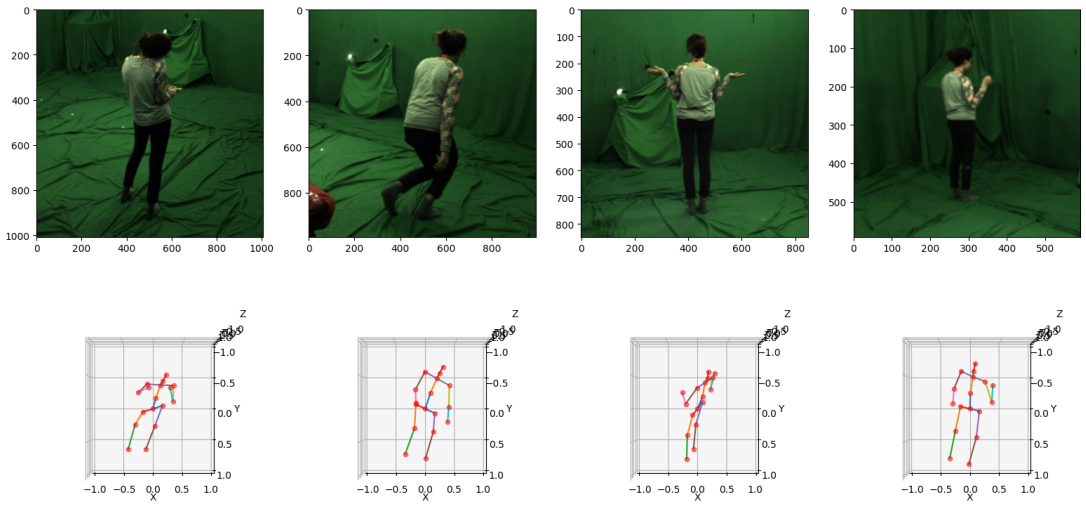


Figure 4.8: Mismatched camera angle

Project human model Inspired by the semi-supervised scheme in [55], we recycle the projected our human model to 2D coordinates and use them in the training loop. The idea is visualized in Figure 4.9, where the path of red arrow if first followed to calculate L. At the mean time, the predicted rotation matrices $\mathbf{R}_i$ are applied to a human model, which is then projected to 2D space using the camera intrinsic matrix given in dataset. The projected 2D keypoints are taken as input by PEBRT (green arrow) to output another set of rotation matrices $\mathbf{R}_i^p$. The loss L_{proj} is calculated and added as a weighted term to the final loss $\mathscr{L}_{total}$. We start by setting α to 0.8, using Equation (4.7).

$$\mathscr{L}_{total} = \alpha L + (1-\alpha) L_{proj} \tag{4.7}$$

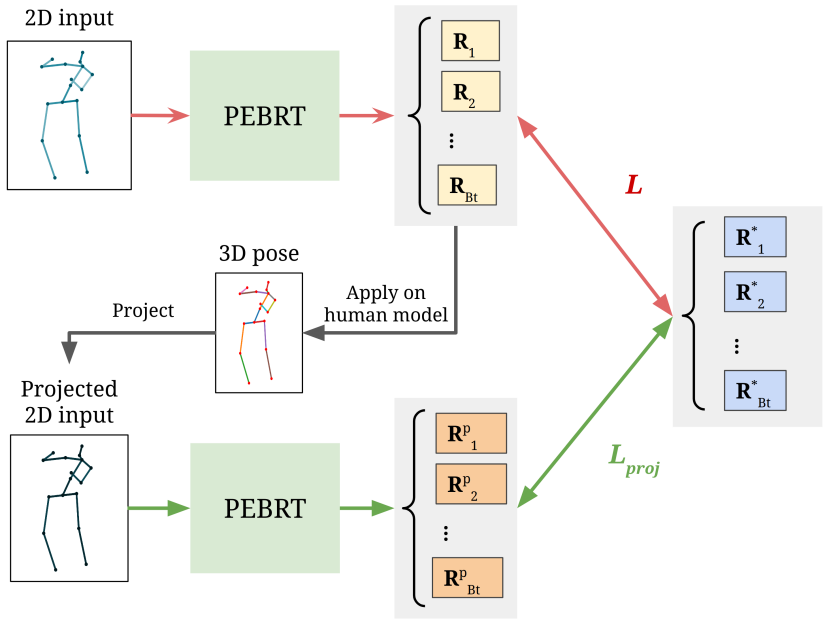


Figure 4.9: Including projected 2D keypoints for training.
(Red arrow) PEBRT processes GT 2D input and outputs predicted rotation matrices $\mathbf{R}_i$. Our human model is applied with $\mathbf{R}_i$ and is projected to 2D pixel coordinates. (Green arrow) The projected 2D keypoints are processed by PEBRT, which yields rotation matrices given projected input $\mathbf{R}_i^p$.

5

Experimental Setup

We evaluate our models on 2 commonly used 3D HPE datasets, Human3.6M [28] and MPI-INF-3DHP [45]. In Human3.6M, our models are trained on 5 subjects (S1, S5, S6, S7, S8) and evaluated on 2 subjects (S9 and S11) on a 17-joint skeleton. 3D reconstruction error is reported in Protocol 1 in MPJPE (mm) for each activity. Whereas in MPI-INF-3DHP, we only demonstrate qualitative results.

5.1. Implementation details

We implement our code in PyTorch, where a single forward pass takes approximately $150ms$ for PETR and $14ms$ for PEBRT on a desktop with Intel i9-9900 CPU and Nvidia RTX 3090 GPU. Both models are trained for 50 epochs on Human3.6M and MPI-INF-3DHP, with a batch size of 2. Table 5.1 gives the approximate number of parameters in both PETR and PEBRT given different number layers of Transformer encoder.

Table 5.1: Number of parameters given different layers of Transformer encoder

# layer(s) of encoder	1	2	4	8
# parameters	150k	300k	590k	1.17M

PETR: Pre-trained weight on COCO2017 [37] for HRNet is used and remains frozen. Learning rate is set to 10^{-4}, learning rate decays by 10^{-5} at a step size of 10, with AdamW [41] being the optimizer.

PEBRT: Learning rate is set to 2×10^{-4}, learning rate decays by 10^{-5} at a step size of 10, with AdamW being the optimizer.

5.2. Experiment results

5.2.1. Discussion on PETR

We conducted ablative experiments to better understand the impact of the number of Transformer encoder layers. In Table 2.1, we present PETR in 1, 2, 4 layers of encoder as well as other contemporary methods benchmarked on Human3.6M.

MPJPE of all versions of PETR fall between that of Zhou *et al.* [84] and Pavlakos *et al.* [52], while having almost twice as much error as state-of-the-art approaches [4, 10, 24]. We reason that the performance is bottlenecked by the heatmap-based architecture of HRNet. To reduce the number of parameters in linear layers, HRNet outputs a lower resolution heatmap. The heatmap resolution is 64x64 in our case, while the input is 256x256. The accuracy of joint detection is hence compromised when mirrored back to the original size of input.

Among the ablation studies, a clear pattern of decreasing error can be observed with increasing number of encoder layers. We consider this as a reasonable result since (1) the mapping takes place in Cartesian coordinates, and (2) the source array has the same order as the target array, i.e. human joint order remains the same.

Table 5.2: MPJPE 1 in mm between the ground-truth 3D joints on Human3.6M for single frame RGB images with ground-truth 2D inputs

MPJPE Protocol 1	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	Walk	WalkD	WalkT	Avg.
Zhou et al. [84] ECCV'16	91.8	102.4	96.7	98.8	113.4	125.2	90.0	93.8	132.2	159.0	107.0	94.4	79.0	126.0	99.0	107.3
Pavlakos et al. [52] CVPR'17	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	59.1	74.9	63.2	71.9
Cai et al. [4] ICCV'19	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	39.2	53.5	41.2	50.6
Ours, PETR (1 layer)	76.6	79.8	92.5	91.0	124.5	111.6	89.5	97.9	117.1	130.7	110.5	89.8	89.7	90.8	99.4	99.4
Ours, PETR (2 layers)	78.8	78.3	96.3	81.0	120.7	115.4	92.9	95.7	113.0	126.5	113.8	87.1	92.0	88.1	97.7	98.5
Ours, PETR (4 layers)	71.4	76.6	92.9	79.3	117.6	113.0	87.3	95.6	112.4	126.1	112.3	87.3	87.9	85.4	93.8	95.9

5.2.2. Discussion on PEBRT

The evaluation of PEBRT is measured in MPBVE (see details in §4.5). We evaluated the works of Martinez *et al.* [44] and Pavllo *et al.* [55] using our novel metrics. The results are shown in Table 5.3. In terms of average performance, PEBRT is outperformed by [44] by 7.4% and by [55] by 15.7%. Action-wise accuracy of PEBRT keeps up with [55] in "Eating" and "Phoning" with less than 5% difference.

In the ablation studies, we see an opposite pattern compared to that in PETR - error increases with increasing layers of encoder. We hypothesize that more layers of Transformer encoder could have led to overfitting. The 4-layer PEBRT model may be overfitting the dataset in this case. However, it cannot be certain without visualizing the learning curve. Another explanation is that source sequence is in joint order, whereas target sequence is in bone order. Although the 4-layer PEBRT might have captured more useful features than its 1-layer version, it is difficult for the model to decipher those information without a decoder.

Table 5.3: MPBVE in mm between the ground-truth 3D joints on Human3.6M for single frame RGB images with ground-truth 2D inputs

MPBVE (mm)	Dir.	Disc.	Eat.	Greet.	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	Walk	WalkD	WalkT	Avg.
Martinez et al. [44] ICCV'17	121.9	137.3	127.4	117.9	137.2	157.5	130.0	115.2	142.8	134.4	149.2	132.1	117.2	154.3	120.4	133.0
Pavllo et al. [55] CVPR'19	103.5	121.6	113.2	112.8	148.1	160.7	118.0	104.4	153.2	160.6	127.2	118.5	92.6	118.3	98.3	123.4
Ours, PEBRT (1 layer)	112.8	134.1	117.0	122.5	145.1	176.3	129.7	125.0	155.3	166.6	154.4	144.6	132.4	185.6	140.0	142.8
Ours, PEBRT (2 layers)	117.2	136.2	115.3	122.7	144.3	186.5	127.7	123.9	156.2	166.5	154.2	141.5	124.6	187.8	142.7	143.1
Ours, PEBRT (4 layers)	119.4	146.1	124.4	124.9	149.0	199.5	131.7	128.6	156.9	164.0	164.1	145.2	126.7	198.5	136.9	147.7

5.2.3. Comparison between PETR and PEBRT

We compare the MPJPE of PETR and PEBRT in Table 5.4. Though benchmarking PEBRT on MPJPE is not objective, it sheds a light on how estimating bone rotation could possibly outperform traditional joint detection approach. The height of human model in this evaluation is set to $1.7m$.

PEBRT constantly outperforms PETR on average MPJPE regardless of number of encoder layers by as much as 15%. We observe PEBRT performing more than 10% better in most activities, especially "Phoning", "SittingDown", and "Smoking". The original dataset in these activities show occluded joints in the majority of frames. The actor was constantly putting his hand in the pocket while phoning and mostly sitting cross-legged on the floor in "SittingDown".

The performance of PEBRT in "WalkingDog", however, falls short of that of PETR by as much as 35%. We attribute this to the fact that the actor frequently turns around in this particular activities. While bone rotation can predict smooth motions, joint detection approach stands out when it comes to more dynamic actions.

Table 5.4: Comparison between PETR and PEBRT on Human3.6M in MPJPE, with the height of human model set to $1.7m$.

MPJPE Protocol 1 (mm)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	Walk	WalkD	WalkT	Avg.
PETR (1 layer)	76.6	79.8	92.5	91.0	124.5	111.6	89.5	97.9	117.1	130.7	110.5	89.8	89.7	90.8	99.4	99.4
PEBRT (1 layer)	74.4	85.1	74.1	80.5	83.0	95.0	75.5	77.5	91.4	93.5	90.0	90.1	86.4	110.6	88.4	86.4
PETR (2 layers)	78.8	78.3	96.3	81.0	120.7	115.4	92.9	95.7	113.0	126.5	113.8	87.1	92.0	88.1	97.7	98.5
PEBRT (2 layers)	76.5	85.5	75.4	81.8	86.9	99.7	76.7	81.2	92.3	95.5	93.0	89.8	88.5	114.1	92.4	88.6
PETR (4 layers)	71.4	76.6	92.9	79.3	117.6	113.0	87.3	95.6	112.4	126.1	112.3	87.3	87.9	85.4	93.8	95.9
PEBRT (4 layers)	76.6	85.7	76.0	79.7	86.8	102.9	77.0	80.8	89.8	93.9	96.9	91.2	89.2	115.3	92.1	88.9

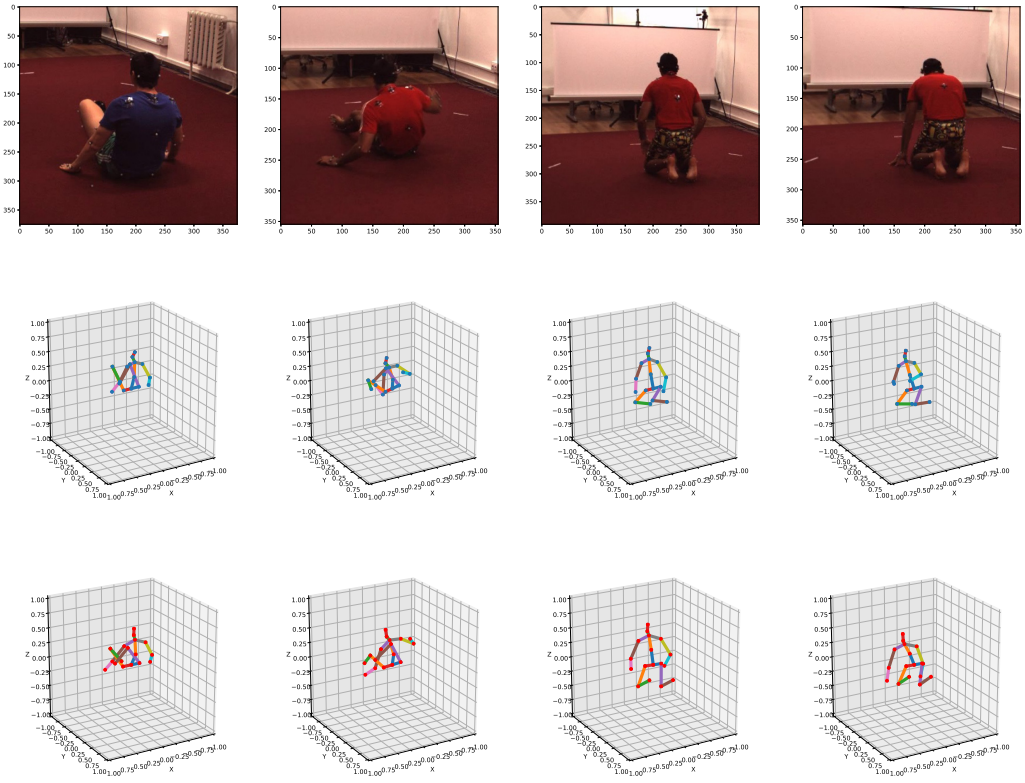


Figure 5.1: Results comparison in "SittingDown" by PETR and PEBRT.
First row: input images; second row: output from PETR; third row: output from PEBRT.

5.2.4. Results with additional loss term

We implement the methods mentioned in §4.6 on 2-layer PEBRT. Result in Table 5.5 shows that their accuracy is outperformed by using only MAEV. In-depth analysis is required to clearly understand why these methods failed to improve the accuracy.

Table 5.5: Additional attempts to improve model accuracy. (PE: position error)

MPJPE Protocol 1 (mm)	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	Walk	WalkD	WalkT	Avg.
PEBRT (2 layers)	**117.2**	**136.2**	**115.3**	**122.7**	**144.3**	**186.5**	**127.7**	**123.9**	**156.2**	166.5	**154.2**	**141.5**	**124.6**	187.8	**142.7**	**143.1**
PEBRT (2 layers) w/ PE	128.0	145.1	124.4	130.0	153.3	192.6	133.8	131.9	167.6	176.5	166.8	150.5	136.1	196.0	152.4	152.4
PEBRT (2 layers) w/ project	122.5	139.1	126.4	126.6	150.8	190.7	134.1	130.6	161.9	**162.3**	166.1	148.6	131.7	**185.0**	145.0	148.1

5.3. Qualitative results

5.3.1. PETR

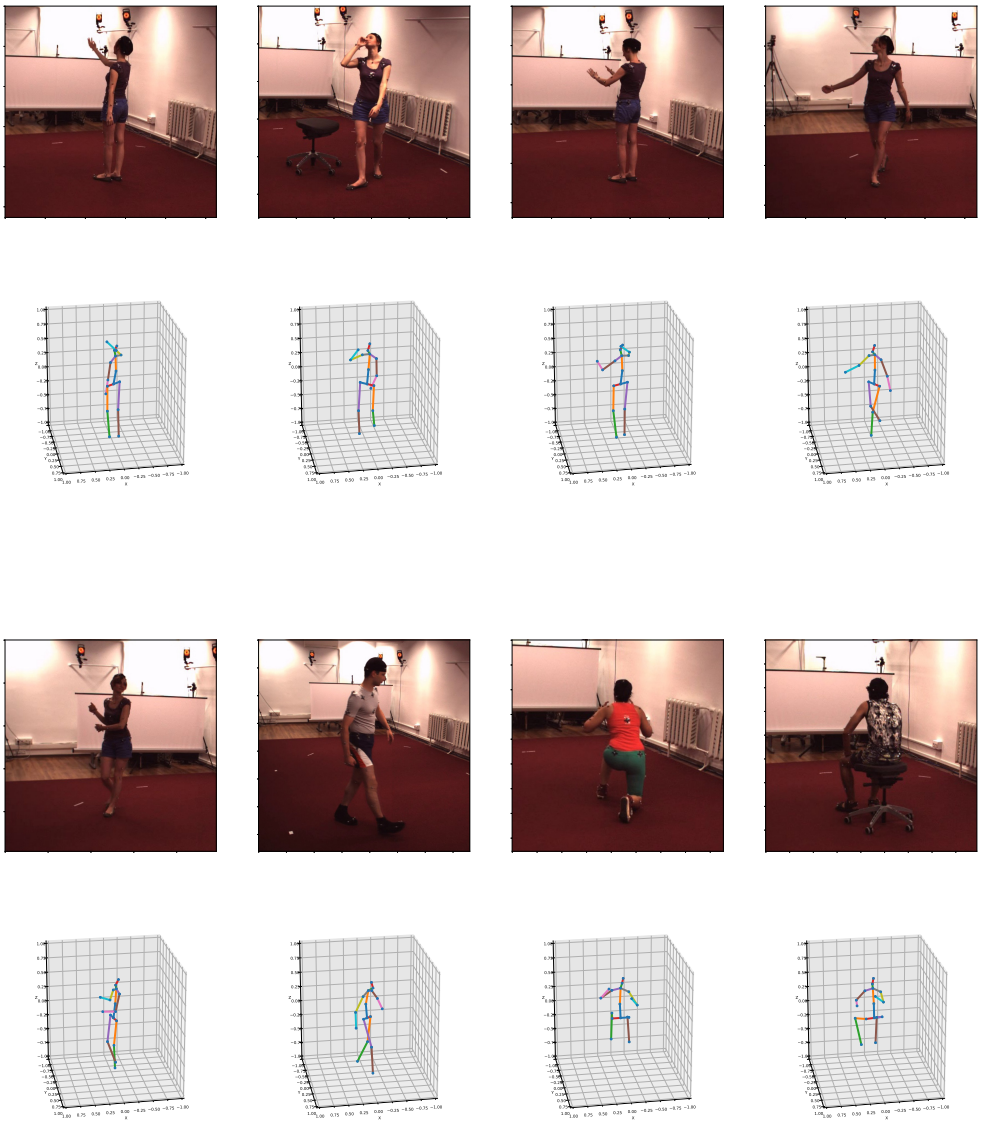


Figure 5.2: Qualitative results on Human3.6M by PETR

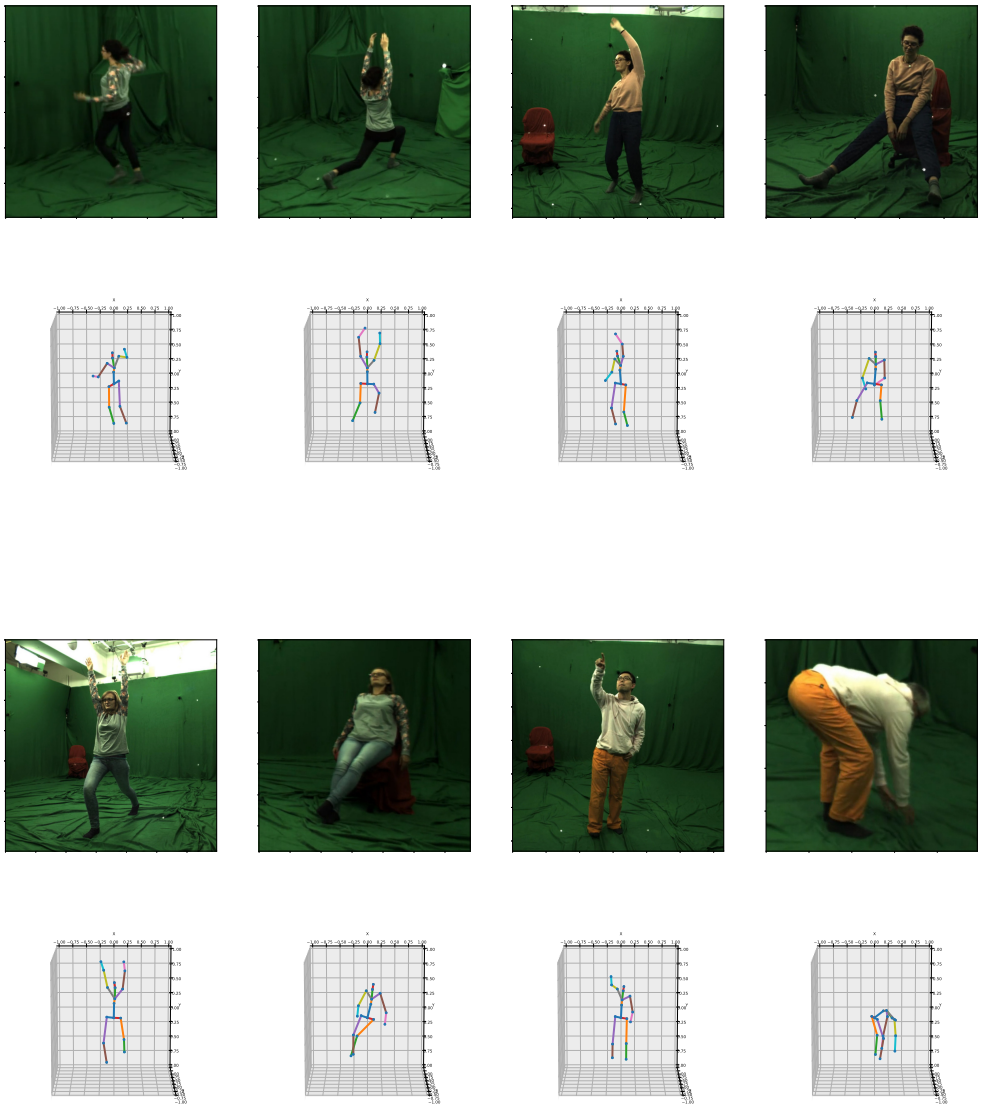


Figure 5.3: Qualitative results on MPI-INF-3DHP by PETR

5.3.2. PEBRT

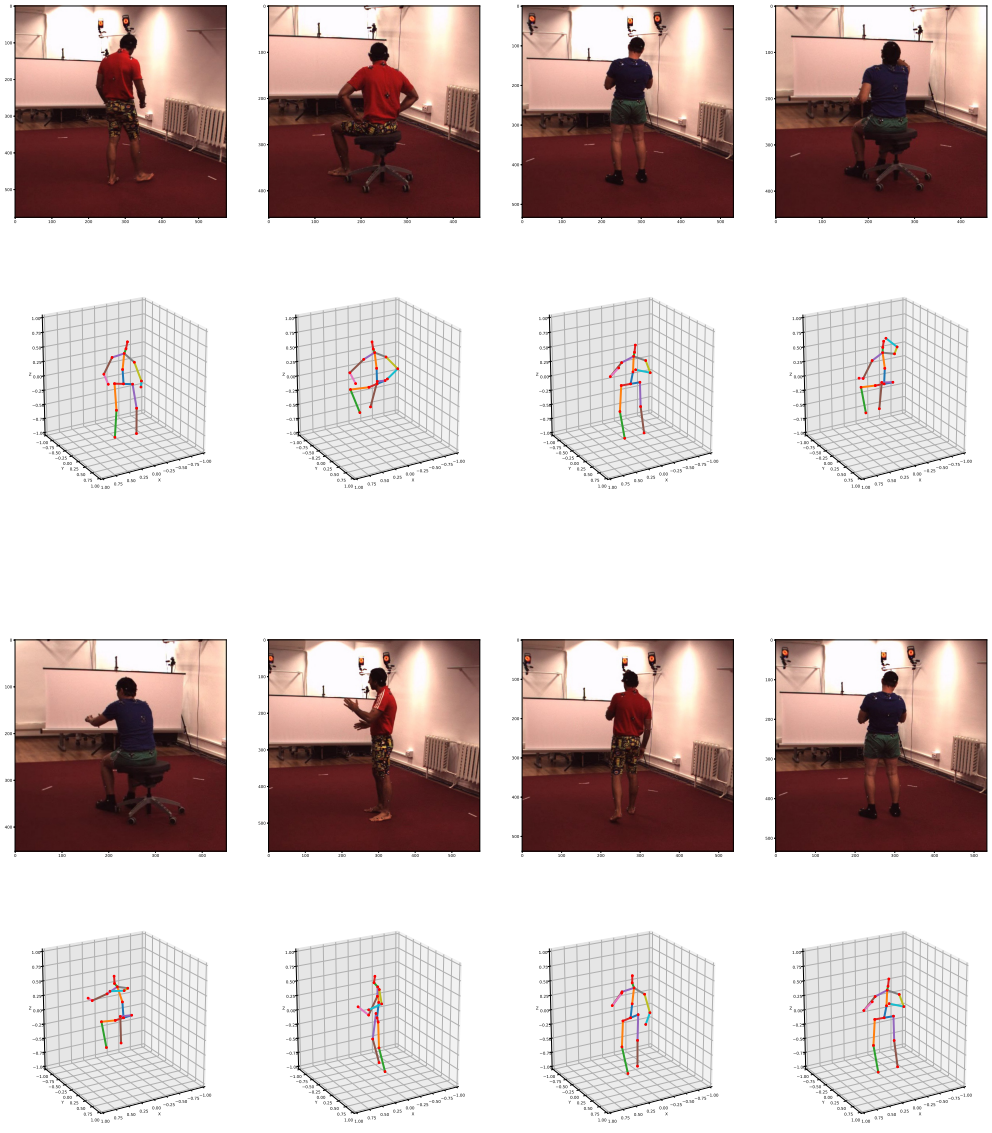


Figure 5.4: Qualitative results on Human3.6M by PEBRT

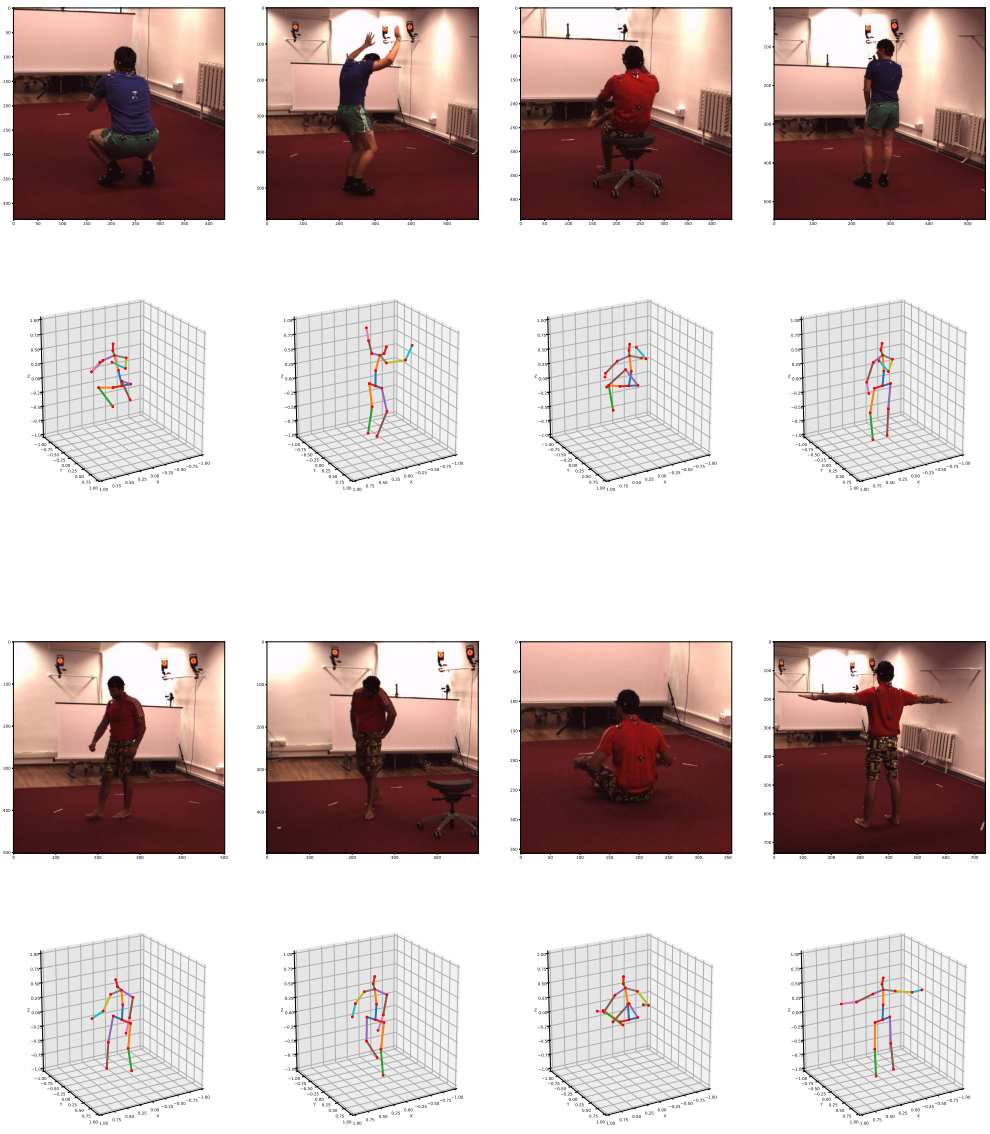


Figure 5.5: Qualitative results on Human3.6M by PEBRT

5.4. Further Improvements

The simplicity and novelty of our approach suggests multiple directions of improvement in future works. For example, adding Transformer decoder is a possible way to improve accuracy but hinders the inference speed and increase the number of parameters.

For PETR, it is possible to use off-the-shelf 2D detectors such as Stacked Hourglass Network (SHN) [49] and Masked R-CNN [26], or open-source API such as Detectron2 [76] and MMPose [14]. While this could potentially yield to higher accuracy, the trade-off is the inference speed retarded by 2D detectors. In addition, HRNet is not finetuned in our experiments.

As for PEBRT, it is possible to integrate with Urdf model and use it in simulation environment such as ROS. Transformer decoder can also be added for potentially better accuracy given the different source and target sequence order.

6

Conclusion

We introduce a deep rotation analysis framework PEBRT based on Transformer encoder for monocular 3D pose estimation. By integrating our human kinematic model, no additional bone length or joint angle constraints are required. This guarantees kinematically realistic human pose and can be extended to applications such as avatar control with a Urdf model in simulation environments. In this work, we propose a new evaluation metric MPBVE that emphasizes 3D pose accuracy regardless of object's body shape, age, or gender. Our framework achieves comparable results to existing methods in Human3.6M. We would like to think of this method as a baseline that tackles 3D human pose estimation from the viewpoint of rotation estimation. We believe future explorations could achieve better accuracy or bring about real-life applications.

Bibliography

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1446–1455. IEEE, 6 2015. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7298751. URL http://ieeexplore.ieee.org/document/7298751/.

[2] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3390–3399. IEEE, June 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00351. URL https://ieeexplore.ieee.org/document/8953699/.

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, P. Gehler, J. Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016.

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 2272–2281. IEEE, 10 2019. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00236. URL https://ieeexplore.ieee.org/document/9009459/.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1302–1310. IEEE, 7 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.143. URL http://ieeexplore.ieee.org/document/8099626/.

[6] Zhiwen Cao, Zongcheng Chu, Dongfang Liu, and Yingjie Chen. A vector-based representation to enhance head pose estimation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, page 1187–1196. IEEE, 1 2021. ISBN 978-1-66540-477-8. doi: 10.1109/WACV48630.2021.00123. URL https://ieeexplore.ieee.org/document/9423257/.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*, volume 12346 of *Lecture Notes in Computer Science*, page 213–229. Springer International Publishing, 2020. ISBN 978-3-030-58451-1. doi: 10.1007/978-3-030-58452-8_13. URL http://link.springer.com/10.1007/978-3-030-58452-8_13.

[8] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation = 2d pose estimation + matching. In *2017 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), page 5759–5767. IEEE, 7 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.610. URL http://ieeexplore.ieee.org/document/8100093/.

[9] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith Mv, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5707–5717. IEEE, 6 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00586. URL https://ieeexplore.ieee.org/document/8953799/.

[10] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *arXiv:2002.10322 [cs]*, 1 2021. URL http://arxiv.org/abs/2002.10322.

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 7103–7112. IEEE, 6 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00742. URL https://ieeexplore.ieee.org/document/8578840/.

[12] Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, 192:102897, 3 2020. ISSN 10773142. doi: 10.1016/j.cviu.2019.102897.

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

[14] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[15] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, S. Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *ECCV*, 2018.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*, 5 2019. URL http://arxiv.org/abs/1810.04805.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[18] R Drillis, R Contini, and M Bluestein. Body segment parameters; a survey of measurement techniques. *Artificial limbs*, 8:44—66, 1964. ISSN 0004-3729. URL http://europepmc.org/abstract/MED/14208177.

[19] A. Elhayek, E. de Aguiar, A. Jain, J. Thompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Marconi—convnet-based marker-less motion capture in outdoor and indoor scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):501–514, 3 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2557779.

[20] Zhijie Fang and Antonio M. López. Is the pedestrian going to cross? answering by 2d pose estimation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1271–1276, 2018. doi: 10.1109/IVS.2018.8500413.

[21] Martin Fisch and Ronald Clark. Orientation keypoints for 6d human pose estimation. *arXiv:2009.04930 [cs]*, 9 2020. URL http://arxiv.org/abs/2009.04930.

[22] Daphne J. Geerse, Bert H. Coolen, and Melvyn Roerdink. Kinematic validation of a multi-kinect v2 instrumented 10-meter walkway for quantitative gait assessments. *PLOS ONE*, 10(10):e0139913, 10 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0139913.

[23] Thomas Golda, Tobias Kalb, Arne Schumann, and Jüergen Beyerer. Human Pose Estimation for Real-World Crowded Scenarios. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019.

[24] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. Poseaug: A differentiable pose augmentation framework for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8575–8584, June 2021.

[25] N. Hamilton, W. Weimar, and K. Luttgens. *Kinesiology: The scientific basis of human motion*. 12th edition, 1971. URL https://accessphysiotherapy.mhmedical.com/book.aspx?bookid=965.

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

[27] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7779–7788, 2020.

[28] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 7 2014. ISSN 0162-8828, 2160-9292. doi: 10.1109/TPAMI.2013.248.

[29] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 7122–7131. IEEE, 6 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00744. URL https://ieeexplore.ieee.org/document/8578842/.

[30] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Found. Trends Comput. Graph. Vis.*, 1, 2005.

[31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, February 2009. ISSN 0920-5691. doi: 10.1007/s11263-008-0152-6. URL https://doi-org.tudelft.idm.oclc.org/10.1007/s11263-008-0152-6.

[32] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22554–22565. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/fec3392b0dc073244d38eba1feb8e6b7-Paper.pdf.

[33] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He. Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 601–604, 2017. doi: 10.1109/ICMEW.2017.8026282.

[34] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1944–1953, June 2021.

[35] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1954–1963, June 2021.

[36] Matthieu Lin, Chuming Li, Xingyuan Bu, Ming Sun, Chen Lin, Junjie Yan, Wanli Ouyang, and Zhidong Deng. Detr for pedestrian detection. *arXiv:2012.06785 [cs]*, 12 2020. URL http://arxiv.org/abs/2012.06785.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, page 740–755. Springer International Publishing, 2014. ISBN 978-3-319-10602-1.

[38] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5063–5072. IEEE, 6 2020. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.00511. URL https://ieeexplore.ieee.org/document/9156272/.

[39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692 [cs]*, 7 2019. URL http://arxiv.org/abs/1907.11692.

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Transactions on Graphics*, 34 (6):1–16, 11 2015. ISSN 0730-0301, 1557-7368. doi: 10.1145/2816795.2818013.

[41] Ilya Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[42] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *2018 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, page 5137–5146. IEEE, 6 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00539. URL https://ieeexplore.ieee.org/document/8578637/.

[43] Wei Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *ArXiv*, abs/2103.15320, 2021.

[44] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, page 2659–2668. IEEE, 10 2017. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.288. URL http://ieeexplore.ieee.org/document/8237550/.

[45] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. *arXiv:1611.09813 [cs]*, 10 2017. URL http://arxiv.org/abs/1611.09813.

[46] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. *arXiv:1712.03453 [cs]*, 12 2017. URL http://arxiv.org/abs/1712.03453.

[47] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4):1–14, 7 2017. ISSN 0730-0301, 1557-7368. doi: 10.1145/3072959.3073596.

[48] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 10132–10141. IEEE, 10 2019. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.01023. URL https://ieeexplore.ieee.org/document/9010999/.

[49] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, page 483–499. Springer International Publishing, 2016. ISBN 978-3-319-46484-8.

[50] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3d human pose estimation by predicting depth on joints. In *2017 IEEE International Conference on Computer Vision (ICCV)*, page 3467–3475. IEEE, 10 2017. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.373. URL http://ieeexplore.ieee.org/document/8237635/.

[51] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv:1802.05751 [cs]*, 6 2018. URL http://arxiv.org/abs/1802.05751.

[52] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *2017 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), page 1263–1272. IEEE, 7 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.139. URL http://ieeexplore.ieee.org/document/8099622/.

[53] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. *arXiv:1805.04092 [cs]*, 5 2018. URL http://arxiv.org/abs/1805.04092.

[55] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 7745–7754. IEEE, 6 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00794. URL https://ieeexplore.ieee.org/document/8954163/.

[56] Ammar Qammaz, Damien Michel, and Antonis A Argyros. A hybrid method for 3d pose estimation of personalized human body models. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2018)*, pages 456–465, Lake Tahoe, NV, USA, 3 2018. IEEE. doi: 10.1109/WACV.2018.00056. URL http://users.ics.forth.gr/argyros/res_personalizedHumanPose.html.

[57] Alec Radford, Jeff Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[58] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv:1906.05909 [cs]*, 6 2019. URL http://arxiv.org/abs/1906.05909.

[59] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. *Reconstructing 3D Human Pose from 2D Image Landmarks*, volume 7575 of *Lecture Notes in Computer Science*, page 573–586. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33764-2. doi: 10.1007/978-3-642-33765-9_41. URL http://link.springer.com/10.1007/978-3-642-33765-9_41.

[60] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2019. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2019.2892985.

[61] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Learning 3-d object orientation from images. In *2009 IEEE International Conference on Robotics and Automation*, pages 794–800, 2009. doi: 10.1109/ROBOT.2009.5152855.

[62] Thomas Schnürer, Stefan Fuchs, Markus Eisenbach, and Horst-Michael Groß. Real-time 4d pose estimation from single depth images. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, page 716–724. SCITEPRESS - Science and Technology Publications,

2019. ISBN 978-989-758-354-4. doi: 10.5220/0007394707160724. URL http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007394707160724.

[63] Gregory G. Slabaugh. Computing euler angles from a rotation matrix, 1999.

[64] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5686–5696. IEEE, 6 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00584. URL https://ieeexplore.ieee.org/document/8953615/.

[65] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 10 2017.

[66] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. *Integral Human Pose Regression*, volume 11210 of *Lecture Notes in Computer Science*, page 536–553. Springer International Publishing, 2018. ISBN 978-3-030-01230-4. doi: 10.1007/978-3-030-01231-1_33. URL http://link.springer.com/10.1007/978-3-030-01231-1_33.

[67] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3961–3970, 2017. doi: 10.1109/ICCV.2017.425.

[68] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv:2012.12877 [cs]*, 12 2020. URL http://arxiv.org/abs/2012.12877.

[69] Shashank Tripathi, Siddhant Ranade, Ambrish Tyagi, and Amit Agrawal. Posenet3d: Learning temporally consistent 3d human pose via knowledge distillation. *arXiv:2003.03473 [cs]*, November 2020. URL http://arxiv.org/abs/2003.03473.

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[71] Marton Veges and Andras Lorincz. Temporal smoothing for 3d human pose estimation and localization for occluded people. *arXiv:2011.00250 [cs]*, 10 2020. URL http://arxiv.org/abs/2011.00250.

[72] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 7774–7783. IEEE, 6 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00797. URL https://ieeexplore.ieee.org/document/8953653/.

[73] Bastian Wandt, Hanno Ackermann, and Bodo Rosenhahn. *A Kinematic Chain Space for Monocular Motion Capture*, volume 11132 of *Lecture Notes in Computer Science*, page 31–47. Springer International Publishing, 2019. ISBN 978-3-030-11017-8. doi: 10.1007/978-3-030-11018-5_4. URL http://link.springer.com/10.1007/978-3-030-11018-5_4.

[74] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao. Robust estimation of 3d human poses from a single image. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, page 2369–2376, 6 2014. doi: 10.1109/CVPR.2014.303.

[75] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021.

[76] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[77] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

[78] Hongfei Xu, Josef van Genabith, Qiuhui Liu, and Deyi Xiong. Probing word translations in the transformer and trading decoder for encoder layers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–85, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.7. URL https://aclanthology.org/2021.naacl-main.7.

[79] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 896–905. IEEE, 6 2020. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.00098. URL https://ieeexplore.ieee.org/document/9157713/.

[80] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv:2012.14214 [cs]*, 12 2020. URL http://arxiv.org/abs/2012.14214.

[81] Biao Zhang, Deyi Xiong, and Jinsong Su. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1166. URL https://aclanthology.org/P18-1166.

[82] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *arXiv:2103.10455 [cs]*, 3 2021. URL http://arxiv.org/abs/2103.10455.

[83] Xiangtao Zheng, Xiumei Chen, and Xiaoqiang Lu. A joint relationship aware neural network for single-image 3d human pose estimation. *IEEE Transactions on Image Processing*, 29:4747–4758, 2020. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2020.2972104.

[84] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, page 186–201. Springer International Publishing, 2016. ISBN 978-3-319-49409-8.

[85] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5738–5746. IEEE, 6 2019. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00589. URL https://ieeexplore.ieee.org/document/8953486/.

[86] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Wolfram Burgard, and Thomas Brox. 3d human pose estimation in rgbd images for robotic task learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, page 1986–1992. IEEE, 5 2018. ISBN 978-1-5386-3081-5. doi: 10.1109/ICRA.2018.8462833. URL https://ieeexplore.ieee.org/document/8462833/.