Master's Thesis of Data Science

Geometry-Aware Data Augmentation for Sequence-to-sequence Multi-Person 3D Pose Estimation

시퀀스 기반 3차원 다인 자세 추정을 위한 기하학적 데이터 증강 기법

February 2023

Graduate School of Data Science
Seoul National University
Data Science Major

Sungchan Park

Geometry-Aware Data Augmentation for Sequence-to-sequence Multi-Person 3D Pose Estimation

Advisor Joonseok Lee

Submitting a master's thesis of
Data Science

December 2022

Graduate School of Data Science
Seoul National University
Data Science Major

Sungchan Park

Confirming the master's thesis written by
Sungchan Park
January 2023

Chair	Taesup Kim	(Seal)
Vice Chair	Joonseok Lee	(Seal)
Examiner	Jay-Yoon Lee	(Seal)

Abstract

3D pose estimation is an invaluable task in computer vision with various practical applications. Recently, a Transformer-based sequence-to-sequence model, MixSTE [60], has been successfully applied to 3D single-person pose estimation by decoupling the 2D-to-3D modeling from pixel-level details. We propose a natural extension of this model from single-person to multi-person problem, adding a novel inter-personal attention for 2D-to-3D lifting. Naturally referring to neighboring frames, this design is highly robust in handling occlusions. However, 3D multi-person pose estimation is still challenging due to extreme data scarcity. From an observation that our 2D-to-3D lifting approach is free from pixel-level details, we propose a novel geometry-aware data augmentation that allows us to infinitely generate diverse training examples from existing single-person trajectories. From extensive experiments on standard benchmarks, we verify that our model and data augmentation method achieve the state-of-the-art, not just on accuracy but also on smoothness. We also qualitatively demonstrate the effectiveness of our approach both on public benchmarks and with in-the-wild videos.

Keyword : 3D, Human Pose, Augmentation, Sequence, Transformer

Student Number : 2021-21537

Table of Contents

Chapter 1. Introduction

3D pose estimation aims to reproduce the 3D coordinates of a person appearing in an untrimmed 2D video. It has been extensively studied in literature with many real-world applications, including sports [4], healthcare [54],games [23], movies [1], and even for an AI-based video compression [51]. Although many applications need fully rendered 3D voxels in the end, under its narrow definition, 3D pose estimation problem treats only a handful number of keypoints in the human body (e.g., neck, knees, or ankles), leaving the recovery of dense voxels as a separate post-processing step.

Depending on the number of subjects, 3D pose estimation is categorized into 3D Single-Person Pose Estimation (3DSPPE) and 3D Multi-Person Pose Estimation(3DMPPE). In this paper, we mainly tackle 3DMPPE, reproducing the 3D coordinates of every person appearing in a video. Unlike extensively studied 3DSPPE, 3DMPPE is still largely uncharted due to two main bottlenecks: occlusion and data scarcity.

First, the occlusion in 3DMPPE is caused by inter-person interactions. Due to the invisible occluded key points, there is unavoidable ambiguity since there are multiple plausible answers for them. Occlusion becomes a lot more severe when a person totally blocks another from the camera, making the model to output inconsistent estimation throughout the frames. Due to the ambiguity,

frame2frame type of models, which take a single frame and produce estimation for each frame at a time, inherently struggle from occlusion.

One way of resolving the occlusion problem is referring to neighboring frames in the video, helping the model to learn lots of cues about the correspondence between keypoints from the neighboring frames. For example, VideoPose3D [41], adopts dilated convolution to attend to neighboring frames, predicting one frame's result from multiple frames at a time (seq2frame approach). MixSTE [60] extends further to seq2seq approach, which takes multiple frames and outputs multiple frames' results at once, taking the Transformer architecture which is widely used for video understanding [3, 32, 46, 58] recently. Particularly, it enjoys a benefit from the 2D-to-3D lifting approach, which learns to map 2D key points detected from an off-the-shelf 2D pose estimation model, to the 3D space.

In this paper, we adopt a similar seq2seq Transformer-based 2D-to-3D structure and propose POTR-3D, naturally extending MixSTE [60] from 3DSPPE to 3DMPPE. Lifting the assumption that there is always a single person in the video, our model tracks up to N people at the same time, introducing an additional self-attention across multiple people appearing in the same frame.

However, although this extension looks straightforward, a naive extension of the seq2seq approach to 3DMPPE suffers from extensive computational cost and lack of training data. Especially, the

data scarcity is a long-standing problem in 3D pose estimation, since collecting 3D annotations requires expensive motion capturing (MoCap) equipment [13]. For this reason, most 3D datasets have a limited number of cameras (e.g., 4 for Human 3.6M [18] and 14 for MPI-INF-3DHP [34]) under limited conditions like the subjects' clothing or lighting. This is the core reason why lots of 3D pose estimators fail for in-the-wild videos.

To tackle this challenge, we introduce a novel 2D-3D pair dataset augmentation strategy. Observing that our 2D-to-3D lifting approach only needs the key points, not the pixel-level details, of the subjects for training, we can easily generate an unlimited number of 2D-3D pairs using given camera parameters under various conditions, e.g., containing arbitrary number of subjects with various occlusion cases. Specifically, we propose four types of novel geometry-aware data augmentation methods, incorporating translation and rotation of the subjects as well as the ground plane. Trained on our augmented data, the proposed model, POTR-3D, significantly improves the quality of 3D multi-person pose estimation, verified by extensive experiments on several benchmarks. We also demonstrate qualitative performance of our model on in-the-wild videos, which have been a long-time challenge for 3DMPPE.

Our contributions can be summarized as follows:

• We propose POTR-3D, a seq2seq 2D-to-3D lifting model for 3DMPPE, which is the first realization of this approach to the best of

our knowledge, being robust on occlusion.

• We devise a simple but effective data augmentation strategy for 3DMPPE, allowing us to generate an unlimited number of augmented datasets and to mitigate the data scarcity problem.

• Our POTR-3D model achieves highly competitive performance on benchmarks and remarkable qualitative results on in-the-wild videos with significant consistency, which is essential for real-world applications.

Chapter 2. Related Work

Human pose estimation has been studied on a single-view (monocular) or on multi-view images. Seeing the scene only from one direction through a monocular camera, the single-view pose estimation is inherently challenging to reproduce the original 3D landscape. Multi-view systems [5, 14, 16, 17, 19, 21, 22, 43, 59] are developed to ease this problem, allowing the model to automatically generate ground truth labels for the single-view counterpart. In this paper, we focus on the single-view 3D human pose estimation, as we are particularly interested in applying it to in-the-wild videos captured without special setups.

2.1. Single-Person 3D Human Pose Estimation

Recent monocular approaches typically adopt neural networks to mitigate the ambiguity of 2D-to-3D joint mapping [8,19,33,37, 38, 40, 42, 49, 52, 55]. Recent surveys [11, 50] provides a comprehensive overview on this task. VideoPose3D [41] performs an effective sequence-Based 2D-to-3D lifting for 3DSPPE using dilated convolution. Recently, Graph Neural Networks (GNNs) are applied to 2D-to-3D lifting [9,31,62]. PoseFormer [64] is a pioneer Transformer-based approach for 3DSPPE, taking the 2D single-person pose sequence of multiple frames. MhFormer [28] generates

multiple hypotheses from 2D single-view of single-person.

2.2. Multi-Person 3D Human Pose Estimation

Top-down approaches first detect individual human in the image, and then estimate location of joints for each detected person [2, 36, 44]. In contrast, bottom-up approaches detect all keypoints in the image, then group them into each appropriate person [26, 39, 57].

Recently, temporal information from video has been exploited to produce more robust predictions by seq2frame methods. Graph Convolution Networks (GCNs) [25] are applied to the task to learn multi-scale features of human and hand poses [6]. These works achieve competent performance, but redundant calculation is known as a common drawback since large amount of sequences are over lapped to infer 3D poses of all frames. On the other hand, sequence-to-sequence (seq2seq) approaches, reconstructing all frames at once, improve the coherence and efficiency of 3D pose estimation. Lin et al. [29] introduces LSTM [15] to estimate 3D poses in a video from a set of 2D key points.

Transformers [48] are widely adopted for 3DMPPE, taking advantage of its strong capability of treating seq2seq problems. TransPose [56] proposes a Transformer-based 2D pose estimation from iamges. PoseFormer [64] constructs a model based on Vision Transformer (ViT) [10] to capture the spatio-temporal dependency sequentially. Strided Transformer [27] reduces the redundancy mentioned above by using strided convolutions. MixSTE [60]

considers motion trajectories of different body joints and applies the seq2seq to better model sequence coherence. Our approach is similar to theirs in applying the Transformer architecture. We consider, however, not only motion trajectories of different body joints but also inter-personal spatial relationships, and apply the seq2seq for better model sequence coherence in multi-person pose estimation.

2.3. Data Augmentation for 3D Pose Estimation

Since 3D pose annotation is expensive to collect, limited training data is an ordinary challenge. 3D multi-person pose data is even more limited. Data augmentation is a well-known method widely used to resolve the training data diversity bottleneck and to improve the generalization ability of the model. PoseAug [13] and AdaptPose [12] address this issue by generating synthetic 3D human motions on the single-person 3D pose estimation problem. Horizontal body flipping is a commonly used for 3DMPPE, but no other methods have been proven to be effective yet, mainly due to lots of physical constraints for augmentation; for instance, all the subjects should be translated on the ground plane, and their occlusions should also be considered.

Chapter 3. Problem Formulation and Notations

3.1. Problem Formulation

In the 3D Multi-person Pose Estimation (3DMPPE) problem, the input consists of a video $\mathbf{V} = [v_1, v_2, \cdots, v_T]$ of T frames, where each frame is $v_t \in \boldsymbol{R}^{H \times W \times 3}$ and (up to) N persons may appear in the video. The task is locating a predefined set of K human body keypoints (e.g., neck, ankles, or knees; see Fig. 4 for an example) in the 3D space for all persons appearing in the video in every frame. The body keypoints in the 2D image space are denoted by $\mathbf{X} \in \boldsymbol{R}^{T \times N \times K \times 2}$, and the output $\mathbf{Y} \in \boldsymbol{R}^{T \times N \times K \times 3}$ specifies the 3D coordinates of each body keypoint for all N people across T frames.

3.2. Notation

For convenience, we define a common notation for 2D and 3D points throughout the paper. Let us denote a 2D point $\mathbf{X}_{t,i,k} \in \boldsymbol{R}^2$ as (u, v), where $\mathrm{u} \in \{0, \cdots, \mathrm{H}-1\}$ and $\mathrm{v} \in \{0, \cdots, \mathrm{W}-1\}$ is the vertical and horizontal coordinate in the image, respectively. Similarly, we denote a 3D point $\mathbf{Y}_{t,i,k} \in \boldsymbol{R}^3$ as $(\mathrm{x}, \mathrm{y}, \mathrm{z})$, where x and y are the coordinates through the two directions parallel to the projected 2D image, and z is the depth from the image.

Chapter 4. The POTR-3D Model

The overall model workflow, depicted in Fig. 1, extends the MixSTE [60] from single-person to multi-person. First, the input frames **V** are converted to a sequence of 2D key points by an off-the-shelf model (Sec. 4.1). Then, they are lifted into the 3D space (Sec. 4.2).

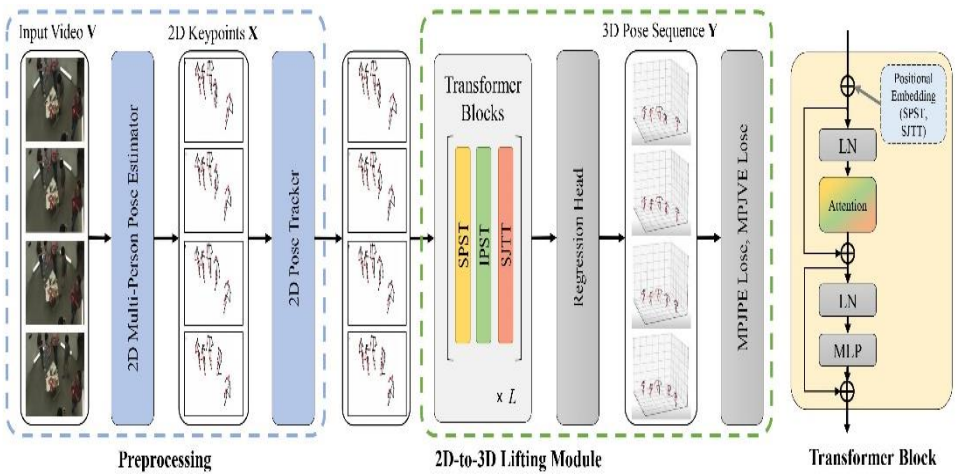

Fig 1 Overview of the POTR-3D. The input video is converted to 2D keypoints, followed by 2D-to-3D lifting, composed of stacked three types of Transformers (SPST, IPST, SJTT).

4.1. Preprocessing

Given an input RGB video $\mathbf{V} \in \boldsymbol{R}^{T\times H\times W\times 3}$, we first need to extract the 2D coordinates $\mathbf{X} \in \boldsymbol{R}^{T\times N\times K\times 2}$ of the N persons appearing in the

video, where $\mathbf{T}$ is the number of frames, and $\mathbf{K}$ is the number of body key points, determined by the dataset. Also, since we treat multiple people in the video, each individual needs to be matched in the input and output. That is, the second index of $\mathbf{X}$ and $\mathbf{Y}$ must be consistent for the same individual across all frames. Any off-the-shelf 2D multi-person pose estimator and a tracking model can be adopted for this preprocessing. In our experiment, we use HRNet [47] and ByteTrack [61] for each, respectively. Note that this preprocessing needs to be done only at testing, since we train our model on augmented videos from a single-person dataset, where the 2D coordinates can be exactly computed from the ground truth and camera parameters (see Sec. 5).

4.2. 2D-to-3D Lifting Module

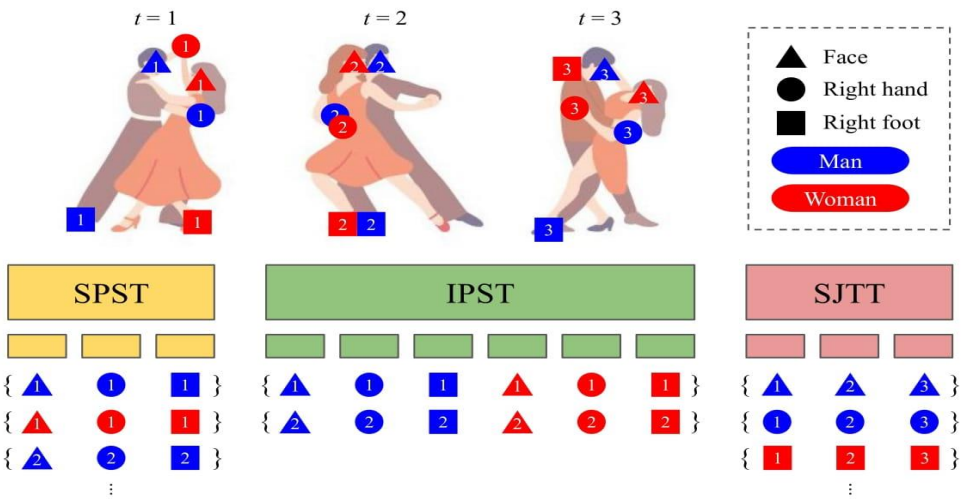

Fig 2 Illustration of the three Transformers in our 2D-to-3D Lifting Module.

Taking $\mathbf{X}$ as its input, this module is to lift it to the 3D coordinates $\mathbf{Y} \in \boldsymbol{R}^{T \times N \times K \times 3}$. To effectively comprehend the spatio-temporal geometric context, we adopt Transformer encoder blocks as our backbone, following MixSTE [60]. Each 2D coordinate $\mathbf{X}_{t,i,k}$, at a specific frame $t \in \{1, \cdots, T\}$ for a specific person $i \in \{1, \cdots, N\}$ and body key point $k \in \{1, \cdots, K\}$, is linearly mapped to a D-dimensional token embedding. Thus, the input is now converted to a sequence of $T \times N \times K$ tokens in $\mathbf{R}^D$, and let us denote these tokens as $\mathbf{Z}^{(0)} \in \boldsymbol{R}^{T \times N \times K \times D}$.

They are fed into the repeated three types of Transformers. Each of them is designed to model a specific relationship between different human body key points: two spatial Transformers modeling intra-person (SPST) and inter-person (IPST) relationships among body keypoints, and a temporal Transformer (SJTT) per each body keypoint across the frames. The role of each Transformer is illustrated in Fig. 2 and detailed below. The input $\mathbf{Z}^{(l-1)} \in \boldsymbol{R}^{T \times N \times K \times D}$ at each layer l goes through the three Transformers in the order of SPST, IPST, and SJTT to contextualize within the sequence, and outputs the same sized tensor $\mathbf{Z}^{(l)} \in \boldsymbol{R}^{T \times N \times K \times D}$.

Single Person Spatial Transformer (SPST). Located at the first stage of each layer l, SPST learns spatial correlation of each person's joints in each frame. Denoting the input to this Transformer as $\boldsymbol{\chi} \in \boldsymbol{R}^{T \times N \times K \times 3}$, SPST takes K tokens of size D corresponding to $\boldsymbol{\chi}_{t,i} \in \boldsymbol{R}^{K \times D}$ for $t \in \{1, \cdots, T\}$ and $i \in \{1, \cdots, N\}$, separately at a time. In other words, SPST takes K different body

keypoints belonging to a same person i at a specific frame t. The output $\mathcal{Y} \in \boldsymbol{R}^{T \times N \times K \times D}$ has the same shape, where each token $\mathcal{Y}_{\boldsymbol{t,i,k}}$ is a transformed one by contextualizing across other tokens belonging to the same person.

The initial input $\boldsymbol{\chi}$ to SPST is $\mathbf{Z}^{(0)} = \mathbf{X}$, extracted by the off-the-shelf models at the very first layer, while at a later layer $l = 2, \cdots, L$ it takes the output from SJTT from the previous layer, $\mathbf{Z}^{(l-1)}$.

Inter-Person Spatial Transformer (IPST). After SPST, IPST learns correlation among body keypoints of every individual in each frame. Through this, the model learns spatial inter-personal relationship in the scene. This is one of the main differences from MixSTE [60]. More formally, this Transformer takes $N \times K$ tokens of size D as input at a time; that is, given the input $\boldsymbol{\chi} \in \boldsymbol{R}^{T \times N \times K \times D}$, all $N \times K$ tokens in the frame $\boldsymbol{\chi_t} \in \boldsymbol{R}^{N \times K \times D}$ are fed into IPST, contextulize from each other, and are transformed to the output tokens $\mathcal{Y}_{\boldsymbol{t}}$. This process is separately performed for $t = 1, \cdots, T$. After IPST, each token is knowledgeable about body keypoints belonging to other people in the same scene, as well as those belonging to the same person.

Single Joint Temporal Transformer (SJTT). The main advantage of simultaneous processing of multiple frames at once is the opportunity to consider global coherence throughout the video. In order to maximize this advantage, the last Transformer SJTT focuses on temporal dynamics of each body keypoint.

Formally, from the input $\boldsymbol{\chi} \in \boldsymbol{R}^{T \times N \times K \times D}$, we create $N \times K$ input

sequences of length T, corresponding to $\boldsymbol{\chi}_{.,i,k} \in \boldsymbol{R}^{T\times D}$ or $i = 1,\cdots,N$ and $k = 1,\cdots,K$. Each sequence is fed into the Transformer, temporally contextualizing each token in the sequence, and the transformed output tokens $\boldsymbol{\mathcal{Y}}_{.,i,k}$ are returned. Completing all $N \times K$ sequences, we have the final output $\boldsymbol{\mathcal{Y}} \in \boldsymbol{R}^{T\times N\times K\times D}$, and this is the output at the l−th layer of our 2D−to−3D lifting module, $\mathbf{Z}^{(l)}$.

These three blocks constitute a single layer of our 2D−to−3D lifting module, and multiple such layers are stacked. A learnable positional encoding is added to each token at the first layer $(l = 1)$ of SPST and SJTT. No positional encoding is added for IPST, since there is no natural ordering between multiple individuals in a video.

Regression Head. After repeating L layers of {SPST, IPST, SJTT}, we get the output tokens for all body keypoints, $\mathbf{Z}^{(L)} \in \boldsymbol{R}^{T\times N\times K\times D}$. This is fed into a regression head, composed of a multilayer perceptron (MLP). It maps each body keypoint embedding in $\mathbf{Z}^{(L)}$ to the corresponding 3D coordinates, $\mathbf{Y} \in \boldsymbol{R}^{T\times N\times K\times 3}$.

4.3. Implementation Details

Depth Normalization. When a 2D image is mapped to the 3D space, depth of each pixel in the 2D image towards the direction of projection needs to be estimated. Following the common practice, we normalize the ground truth depth z by the focal length.

Root Joints. Among the body key points of a person I at frame t, one is chosen as a root joint (typically the body center; denoted by

$\mathbf{Y}_{t,i,1} = (x, y, z) \in \boldsymbol{R}^3$). The ground truth is given by $(x, y, \bar{z})$, where (x, y) is 2D coordinate of the root joint and $\bar{z}$ is its normalized depth. For root joints, the model learns the absolute values, $(x, y, \bar{z})$. Other regular joints, $\mathbf{Y}_{t,i,k} = (x, y, z) \in \boldsymbol{R}^3$ with $\mathrm{k} = 2, \cdots, \mathrm{K}$, are represented as the relative difference from the root joint of the person, $\mathbf{Y}_{t,i,1}$.

4.4. Training Objectives

Given the predicted $\widehat{\mathbf{Y}} \in \boldsymbol{R}^{T \times N \times K \times 3}$ and ground truth $\mathbf{Y} \in \boldsymbol{R}^{T \times N \times K \times 3}$, we minimize the following two losses.

Mean per Joint Position Error (MPJPE) Loss is the L2 distance loss between the prediction and the target:

$$\boldsymbol{L}_{MPJPE} = \frac{1}{N_V TNK} \sum_{n=1}^{N_V} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{k=1}^{K} L2Dist\left(\widehat{\mathbf{Y}}^{(n)}{}_{t,i,k}, \mathbf{Y}^{(n)}{}_{t,i,k}\right)$$

where $\widehat{\mathbf{Y}}^{(n)}$ and $\mathbf{Y}^{(n)}$ are the predicted and true coordinates of the $\mathrm{n}-$th example in the test set with N_V videos.

Mean per Joint Velocity Error (MPJVE) Loss [41] is the L2 distance of the first derivative of MPJPE, measuring smoothness of the predicted sequence.

$$\boldsymbol{L}_{MPJVE} = \frac{1}{N_V TNK} \sum_{n=1}^{N_V} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{k=1}^{K} L2Dist\left(\frac{\partial \widehat{\mathbf{Y}}^{(n)}{}_{t,i,k}}{\partial t}, \frac{\partial \mathbf{Y}^{(n)}{}_{t,i,k}}{\partial t}\right)$$

The overall loss $\boldsymbol{L}$ is given by a weighted sum of the two losses;

that is, $\boldsymbol{L} = \boldsymbol{L}_{MPJPE} + \lambda * \boldsymbol{L}_{MPJVE}$, where λ controls the relative importance between them. Optionally, different weights can be applied to the root joints and others.

Chapter 5. Geometry-Aware Data Augmentation

Thanks to the 2D-to-3D lifting approach, we treat only the coordinates of body keypoints either in 2D or 3D. Thus, the pixel-level details are naturally decoupled from their canonical location. Being free from the pixel-level details, we can freely augment the training data as proposed below, helping us to resolve the data scarcity issue for this task.

Specifically, we take N samples $(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})$ captured by a fixed camera from a single-person dataset, where $\mathbf{X}^{(i)} \in \boldsymbol{R}^{T\times K\times 2}$, $\mathbf{Y}^{(i)} \in \boldsymbol{R}^{T\times K\times 3}$, and $\mathrm{i} = 1, \cdots, \mathrm{N}$. We may simply overlay them onto a single video, producing $\mathbf{X} \in \boldsymbol{R}^{T\times N\times K\times 2}$, $\mathbf{Y} \in \boldsymbol{R}^{T\times N\times K\times 3}$, respectively. This $(\mathbf{X}, \mathbf{Y})$ is an augmented 3DMPPE training example, and repeating this process with different combinations of samples will infinitely create new 3DMPPE examples.

Furthermore, we consider additional data augmentation on the trajectories, e.g., randomly translating or rotating them, to introduce additional randomness and fully take advantage of existing data. However, there are a few additional factors to consider: the ground plane, potential occlusion, and feasibility of the augmented trajectories.

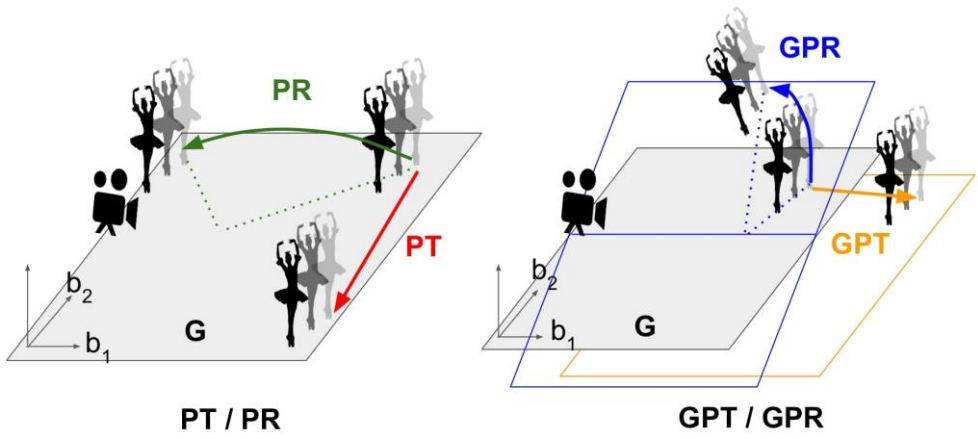

Fig 3 Illustration of the proposed data augmentation methods.

5.1. Ground Plane Orientation

Although translating or rotating a trajectory in 3D space sounds trivial, most natural scenes do not fully use the three degree of freedom, because of an obvious fact that people usually stand on the ground. Geometrically, subjects in a video share the common ground plane, with a few exceptions like a swimming video. As feet generally touches the ground, we estimate the ground plane by collecting feet coordinates from all frames captured by a fixed video and fit them with a linear regression model, producing a 2D linear manifold $\mathbf{G}$ within the 3D space. We choose its two basis vectors, $\{b_1, b_2\}$, perpendicular to the normal vector of $\mathbf{G}$. We propose four types of data augmentation in Fig. 3. By combining these, we generate abundant sequences mimicking various multi-person and camera movements.:

• Person Translation (PT): The target person is translated randomly along the basis $\{b_1, b_2\}$ on the ground plane. We sample the amount of displacement $\Delta\alpha$, $\Delta\beta$ on $\{b_1, b_2\}$ from a Gaussian distribution $N(0, \sigma)$, where σ is a hyper−parameter. Each individual moves by different amount.

• Person Rotation (PR): We sample an angle ω uniformly within $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. The subject is rotated by the same angle ω across the entire sequence to preserve natural movement, with respect to the normal vector of the ground plane about the origin at the mean of all keypoints.

• Ground Plane Translation (GPT): The entire ground plane is shifted through the depth (z) axis by a randomly chosen distance among $\{-1.0, 0.0, 1.5, 3.0\}$ meters, towards (negative) or away from the camera. As GPT is applied to the ground plane, it affects all subjects homogeneously.

• Ground Plane Rotation (GPR): The entire ground plane is rotated by an angle randomly chosen among $\left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$, with respect to the basis b_1, whose direction is more parallel to the x−direction of camera. This is to generate vertically diverse views, which are challenging to create with the other 3 augmentations.

5.2. Handling Occlusions

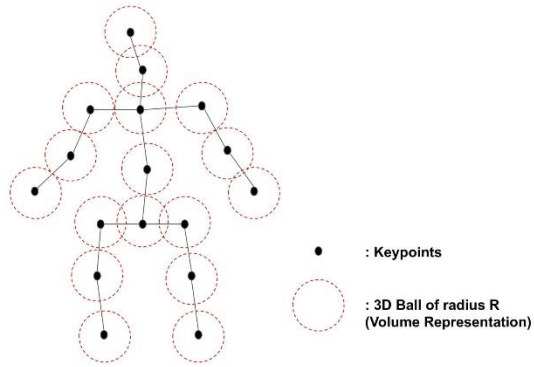

Fig 4 Simple Volume Representation of Person. A volume representation of person needed for generating occlusion.

As multiple subjects in a scene are projected to 2D, they may occlude each other. In such cases, the occluded body parts should not be included in the output. At a glance, this looks trivial; we compute the distance between each keypoint and the camera, and if two or more points are on the same ray from the camera, we leave only the closest one.

Since the human body has some volume, however, two body parts (either from the same person or from different ones) may occlude if the two key points are projected close enough, even though they do not exactly coincide. From this observation, we propose a simple volume representation of person, illustrated in Fig. 4. The volume of each body part is modeled as a 3D-ball centered at the corresponding keypoint. Once projected to the 2D plane, the circles are considered to overlap if the distance between the two circles' centers is shorter than the larger one's radius. Then, the one with the larger depth is

occluded. Optionally, the occluded keypoints may be slightly perturbed, since the keypoint location is not exactly precise anyway.

5.3. Feasibility Constraints

Once the augmented trajectories are generated in the 3D space, we need to project them to the 2D image coordinate to make them paired as a training example. This is done by applying the pinhole camera model. A point $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ in the 3D space is mapped to (u, v) by

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

where f_u, f_v are the focal lengths, and c_u, c_v are the center location in the 2D image coordinates.

Lastly, we need to check if they are feasible as a training example. First of all, the depth z of all target 3D keypoints should be positive. Otherwise, a subject with negative depth will appear flipped both vertically and horizontally, located behind the camera in the 3D space.

Also, the resulting trajectory should be entirely located within the 2D frames. Precisely, we keep the root key points to appear within the image boundary and let other joints potentially be out of the scene. For this, we might naively filter out examples that violate the constraints and regenerate, but this is not efficient. Instead, we apply PR, GPT, and GPR first, and PT at the last. Unlike other operations, we can constrain the feasible range for PT individually,

satisfied simply by solving a constrained linear programming:

$$0 \leq f_u \frac{x + \Delta x}{z + \Delta z} + c_u < W, \;\; 0 \leq f_v \frac{y + \Delta y}{z + \Delta z} + c_v < H, \;\; 0 \leq z + \Delta z$$

where, $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ is an original root joint in the 3D space, $(\Delta \mathrm{x}, \Delta \mathrm{y}, \Delta \mathrm{z})$ is the amount of displacement applied to this subject, converted from $(\Delta\alpha, \Delta\beta)$ on the basis $\{b_1, b_2\}$ to the standard basis (e_1, e_2, e_3), and W, H is the width and height of the image.

Chapter 6. Experiments

6.1. Experimental Settings

Datasets. MuPoTS-3D [35] is one of the most representative datasets for the monocular 3DMPPE. It consists of 20 a few seconds long sequences with 2-3 people interacting with each other. Since this data is made only for evaluation purpose, MuCo-3DHP [35] is widely paired with it for training. MuCo-3DHP was artificially composited from a dataset MPI-INF-3DHP [34], which contains 8 subjects' various motions captured from 14 different cameras.

CMU Panoptic [21] is another popular 3D multi-person dataset, mainly used for multi-view settings. It contains 60 hours of video with 3D poses and tracking information captured by multiple cameras. Following [2], we use video sequences of camera 16 and 30 for both training and testing. This training set consists of sequences with 3 activities, Haggling, Mafia, and Ultimatum, and the test set consist of sequences with an additional activity, Pizza.

We train our model on the synthesized training set using the proposed augmentation method in Sec. 5. We use MPI-INF-3DHP as the source of augmentation for MuPoTS-3D experiment. For CMU-Panoptic, we augment on its training partition.

Evaluation Metrics. First, we measure Percentage of Correct Keypoints (PCK). Given a threshold τ, a keypoint prediction is

regarded correct if the L2 distance between the predicted and true points in the 3D space is within the threshold. We report PCK metrics with (PCKrel) and without (PCKabs) the root alignment, following the convention. We use the common setting of τ = 150mm. Higher PCK indicates better performance.

MPJPE is the mean L2 distance between prediction and ground truth, used for our evaluation on CMU-Panoptic [21]. MPJVE [41] measures the smoothness or consistency of each keypoint's flow over time by the average of the L2 distance between the first derivatives of the predicted and true keypoints. High MPJVE indicates more jitterings between frames, making impractical to apply the method in practice. Lower MPJPE and MPJVE indicate better performance. And they are calculated exactly the same with the MPJPE loss and MPJVE loss.

Competing Models. We compare our POTR-3D method against six baseline models: VirtualPose [45], SingleStage [39], SMAP [63], SDMPPE [36], TDBU-Net [7], and MubyNet [57].

Implementation Details. The input 2D poses are obtained by fine-tuning HRNet-W48 [47]. As it operates in a frame2frame manner, we track each individuals (i.e., stitching) over the whole frames of the input video. We use ByteTrack [61] for tracking, merging with the appearance gallery idea [53] to consider appearance variation caused by movements. While tracking individuals frame by frame, the most recent 100 appearance features are stored in their tracklet. Note that these off-the-shlef models are used only at testing, not at

training, since we train on synthetic data we augment. We use Adam optimizer [24] with batch size of 16, dropout rate of 0.1, and adopt GELU activation function.

6.2. Quantitative Comparison

Method	PCKrel (%) ↑	PCKabs (%) ↑	MPJVErel (mm) ↓	MPJVEabs (mm) ↓
From estimated 2D keypoints (No GT Used)				
VirtualPose[45]	–	44.0	–	–
SingleStage[20]	80.9	39.3	–	–
SMAP[63]	73.5	35.2	–	–
3DMPPE[36]	81.8	35.2	25.8	120.4
POTR-3D (Ours)	**83.7**	**50.9**	**10.8**	**16.3**
From 2D ground truth keypoints				
TDBU-Net[7]	89.6	–	26.3	
POTR-3D (Ours)	**91.0**	33.7	**7.9**	10.2

Tab 1 Quantitative Comparison on MuPoTS-3D. The best scores are marked in boldface.

MuPoTS3D. According to Tab. 1, ours achieves the highest PCKrel and PCKabs with 1.9% and 6.9% gain, respectively, from the previous state-of-the-art. Here we the augmentation consists of PT and PR makes the best performance. Generally POTR-3Doutperforms others at most sequences, furthermore we also emphasize that ours particularly outperforms baselines on sequences with some sequences with heavy occlusions (e.g., TS3, TS14, TS20),

and with some unusual distance from camera (e.g., TS6, TS13), as reported in Tab. 2, and Tab. 3. It indicates the effectiveness of our seq2seq modeling to solve occlusion, and the benefits of augmentation which can make a variety of distance.

Method	PCKrel (%) ↑	TS1	TS2	TS3	TS4	TS5	TS6	TS7
SingleStage[20]	80.9	–	–	–	–	–	–	–
SMAP[63]	73.5	88.8	71.2	77.4	77.7	80.6	49.9	86.6
3DMPPE[36]	81.8	**94.4**	77.5	79.0	81.9	85.3	72.8	81.9
POTR-3D (Ours)	**83.7**	92.0	**80.2**	**93.7**	**84.0**	**85.4**	**75.1**	**91.5**
Method	**TS8**	**TS9**	**TS10**	**TS11**	**TS12**	**TS13**	**TS14**	**TS15**
SingleStage[20]	–	–	–	–	–	–	–	–
SMAP[63]	51.3	70.3	89.2	72.3	81.7	63.6	44.8	79.7
3DMPPE[36]	**75.7**	**90.2**	**90.4**	79.2	79.9	75.1	72.7	81.1
POTR-3D (Ours)	74.3	70.7	88.4	**85.6**	**86.5**	**83.1**	**77.1**	**82.8**
Method	**TS16**	**TS17**	**TS18**	**TS19**	**TS20**			
SingleStage[20]	–	–	–	–	–			
SMAP[63]	86.9	81.0	75.2	73.6	67.2			
3DMPPE[36]	89.9	**89.6**	81.8	81.7	76.2			
POTR-3D (Ours)	**90.8**	86.8	**87.5**	**85.7**	**82.6**			

Tab 2 Quantitative Comparison(PCKrel) on MuPoTS-3D for Individual Test Videos. The best scores are marked in boldface.

Method	PCKabs (%) ↑	TS1	TS2	TS3	TS4	TS5	TS6	TS7
VirtualPose[45]	44.0	–	–	–	–	–	–	–
SingleStage[20]	39.3	–	–	–	–	–	–	–
SMAP[63]	35.2	21.4	22.7	58.3	27.5	37.3	12.2	49.2
3DMPPE[36]	31.5	**59.5**	**44.7**	51.4	46.0	52.2	27.4	23.7
POTR-3D (Ours)	**50.9**	50.1	42.1	**71.0**	**60.5**	**58.6**	**50.4**	**66.9**

Method	TS8	TS9	TS10	TS11	TS12	TS13	TS14	TS15
VirtualPose[45]	–	–	–	–	–	–	–	–
SingleStage[20]	–	–	–	–	–	–	–	–
SMAP[63]	40.8	**53.1**	43.9	**43.2**	43.6	39.7	28.3	49.5
3DMPPE[36]	26.4	39.1	23.6	18.3	14.9	38.2	26.5	36.8
POTR-3D (Ours)	**41.5**	50.0	**69.6**	42.3	**49.2**	**63.2**	**49.3**	**69.0**

Method	TS16	TS17	TS18	TS19	TS20
VirtualPose[45]	–	–	–	–	–
SingleStage[20]	–	–	–	–	–
SMAP[63]	23.8	18.0	26.9	25.0	38.8
3DMPPE[36]	23.4	14.4	19.7	18.8	25.1
POTR-3D (Ours)	**35.6**	**36.9**	**35.3**	**29.3**	**46.3**

Tab 3 Quantitative Comparison(PCKabs) on MuPoTS-3D for Individual Test Videos. The best scores are marked in boldface.

PT	PR	GPT	GPT	Size	PCKrel (%) ↑	PCKabs (%) ↑	MPJVErel (mm) ↓	MPJVEabs (mm) ↓
				0.4M	80.2	37.8	10.9	17.0
V				0.7M	82.8	40.6	11.4	18.7
				1.3M	82.8	40.9	10.9	16.8
				0.4M	81.0	41.3	11.0	16.7
V	V			0.7M	83.7	**50.9**	10.9	**16.3**
				1.3M	83.0	47.1	10.8	18.1
				0.4M	82.3	42.1	10.9	17.1
V	V	V		0.7M	82.2	45.9	**10.7**	17.0
				1.3M	83.7	45.4	**10.7**	18.7
				0.4M	81.3	41.9	11.2	17.1
V	V	V	V	0.7M	83.3	48.1	10.8	18.3
				1.3M	**84.3**	46.1	11.0	16.5

Tab 4 Ablation Study on Augmentation Strategy on MuPoTS-3D. The best scores are marked in boldface.

Method	MPJPErel (mm) ↓					MPJVErel (mm) ↓
Sequence	*Haggling*	*Mafia*	*Ultimatum*	*Pizza*	Avg.	Avg.
VirtualPose[45]	**54.1**	61.6	**54.6**	65.4	58.9	–
SMAP[63]	63.1	60.3	56.6	67.1	61.8	–
MubyNet[57]	72.4	78.8	66.8	94.3	78.1	–
3DMPPE[36]	89.6	91.3	79.6	90.1	87.7	–
POTR-3D (Ours)	59.8	**57.0**	56.6	**59.7**	**58.3**	**4.6**
POTR-3D (Ours; GT)	54.8	39.0	43.1	40.6	44.4	3.4

Tab 5 Quantitative Comparison on CMU Panoptic. The models are trained on {Haggling, Mafia, Ultimatum}, and generalized to Pizza. The best scores are marked in boldface.

We also evaluate POTR-3D with 2D ground truth (GT) keypoints (instead of those by HRNet [47]) to see the upper bound, reported in Tab. 1. We observe a significantly higher PCKrel than the regular experiment, outperforming previous best scores on GT [7]. This indicates that the performance of POTR-3D is highly limited by the 2D keypoint detector, not the proposed model itself.

As MPJVE has not been reported in previous works, we report it in Tab. 1 only for methods open-sourced. POTR-3D significantly improves the MPJVE metrics, compared to previous frame2frame [36] and seq2seq [7] methods. By nature, frame2frame methods are unable to optimize MPJVE, as they do not treat temporal information. Also, as we directly optimize an MPJVE loss term, the huge gap between ours and frame2frame methods is inevitable. This means, however, this practically important MPJVE metric has been overlooked, allowing severe jittering that are often observed with frame2frame approaches. In contrast, our model achieves even stronger MPJVE metrics than the other seq2seq baseline [7] on the 2D GT keypoints, proving the improved smoothness of our model.

CMU-Panoptic. POTR-3D also achieves the state-of-the-art performance on CMU-Panoptic, 0.6mm leading the previous state-of-the-art [45]. In contrast to MuPoTS-3D, CMU-Panoptic contains videos with a denser crowd of 3-8 subjects, making the tracking step more challenging. The result indicates that POTR-3D operates well even in this challenging situation. However, as the same camera setting is used for both training and testing, the

challenge becomes a bit relaxed compared to MuPoTS-3D. For this reason, POTR-3D trained with data augmented only by PT, PR leads to the best performance. Also, POTR-3D achieves significantly higher performance than others on the Pizza sequence unseen at training, with 5.7mm gain. This verifies generalizability of POTR-3D.

6.3. Ablation Study

We further investigate the best data augmentation strategy proposed in Sec. 5, specifically, what kind of operations benefit the most and how many examples are needed. We compare 4 different combinations of the proposed methods (PT, PT+PR, PT+PR+GPT, and PT+PR+GPT+GPR) with 3 sizes (0.4M, 0.7M, and 1.3M samples).

PT	PR	GPT	GPR	Size	MPJPErel (mm) ↓	MPJVErel (mm) ↓	APrel @150mm ↑
V				0.5M	87.1	4.5	65.2
V	V			0.5M	**58.3**	4.6	79.8
V	V	V		0.5M	58.5	5.0	65.5
V	V	V	V	0.5M	72.7	4.8	70.3
V	V	V	V	0.8M	68.8	**4.2**	**83.7**

Tab 6 Ablation Study on Augmentation Strategy on CMU-Panoptic. The best scores are marked in boldface.

PT	PR	GPT	GPR	Size	MPJPErel (mm) ↓	MPJVErel (mm) ↓	APrel @150mm ↑
V				0.5M	142.9	3.1	39.5
V	V			0.5M	136.8	6.1	31.5
V	V	V		0.5M	138.2	5.3	80.4
V	V	V	V	0.5M	67.2	3.7	73.7
V	V	V	V	0.8M	**58.4**	**2.7**	**81.4**

Tab 7 Ablation Study on Camera Setting of CMU-Panoptic. The best scores are marked in boldface.

Tab. 4 shows the performance on MuPoTS-3D. First of all, larger size generally benefits, as expected. Without a limit, the proposed data augmentation may further improve the result with a larger training set. Among the combinations, using more variety of operations generally helps, where the largest one with all operations achieves the best PCKrel. A similar experiment is conducted on CMU-Panoptic, summarized in Tab. 6. Similarly, a larger setup using all operations leads to superior performance in general.

For the experiment on CMU-Panoptic in Sec. 6.2, the conventional benchmark uses the same cameras (camera 16, 30) for both training and test sets. Thus, the model is hard to fully enjoy the benefit of our ground plane augmentation (GPR / GPT). Here, we further evaluate on videos of Haggling, Ultimatum captured by different cameras (camera 6, 13) from the ones used for training Fig. 5 illustrates each camera's view point, and Tab. 7 shows the result. Notations are following Sec. 6.2.

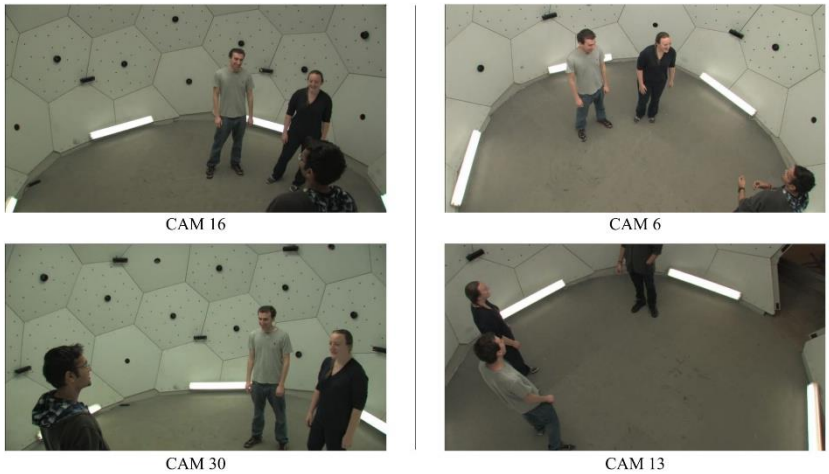

Fig 5 Cameras' view points of CMU-Panoptic. (Left) Cameras used in conventional benchmark for both training and testing. **(Right)** Cameras used for testing in Sec. 6.3.

We confirm that the full augmentations significantly outperform others that care less about the camera view point. Also, we observe a larger augmented dataset benefits more. Furthermore, the best option (last row) achieves competitive performance comparing to the testing performance of Tab. 6, which uses same camera setting with training. This is notable because we do not use any clue about the camera 6 and 13. This proves that GPR benefits the augmentation process to be robust to camera view changes, aligning with our expectation, and indicates proper GPT/GPR leads the model to better generalize.

6.4. Qualitative Results

In addition to the benchmark datasets, we evaluate POTR-3D on a lot more challenging in-the-wild scenarios, e.g., group dancing video or figure skating. Fig. 6 demonstrates the performance of our model on a few examples of in-the-wild videos. In spite of occlusions, we see that POTR-3D precisely estimates poses of multiple people. To the best of our knowledge, this is the first work to present such accurate and consistent 3DMPPE results on in-the-wild videos. The actual video is available on https://github.com/POTR3D/CVPR2023.

Fig. 7 shows a couple of failure cases of our method. When people are not standing or when feet are not shown within the video frame, our method fails to detect the ground plane correctly, leading to incorrect pose estimation. Another challenge is when the off-the-shelf tracking method fails. When it misses a subject, we see that our method cannot detect the pose.

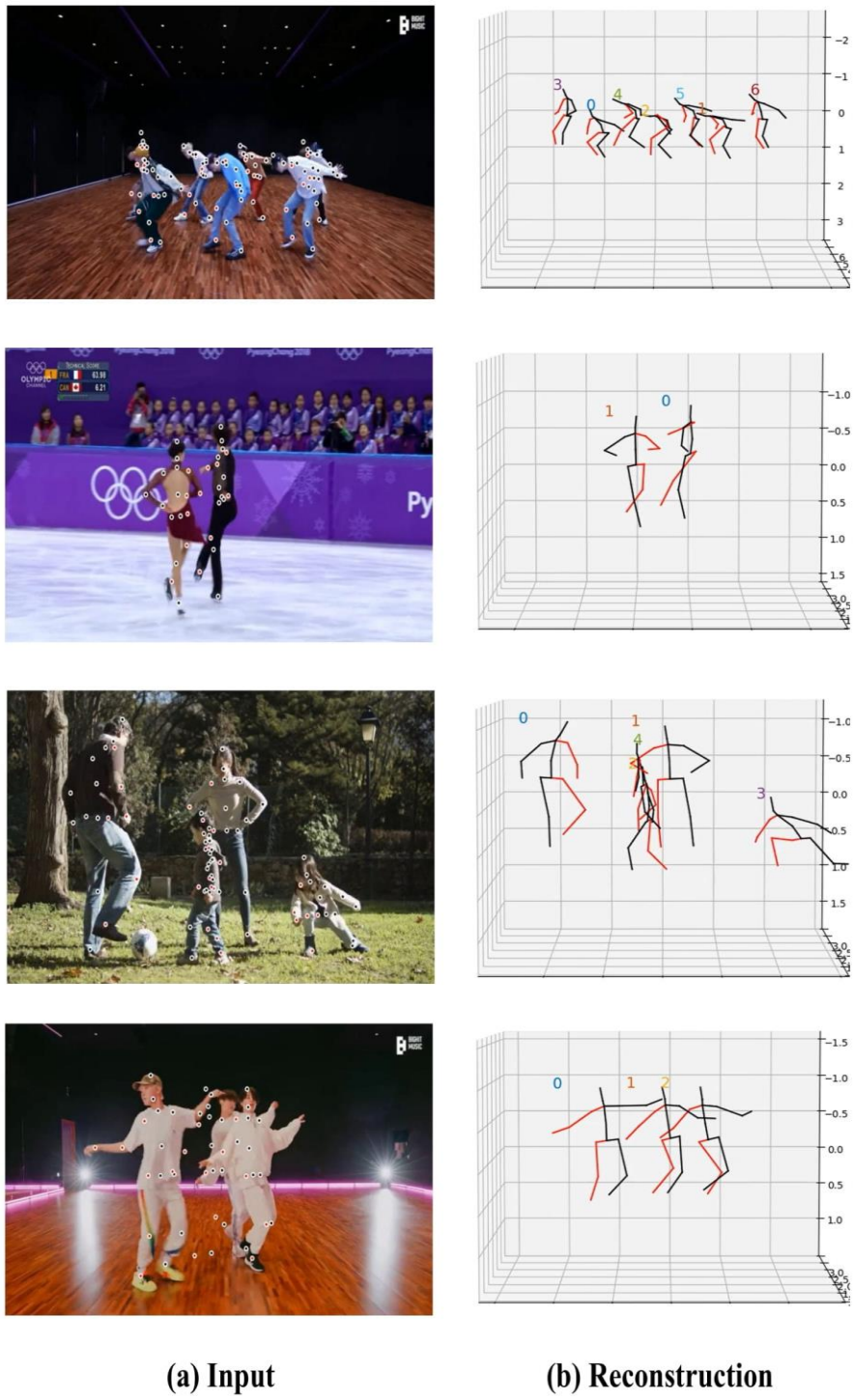

Fig 6 Demonstration of POTR-3D on in-the-wild videos.

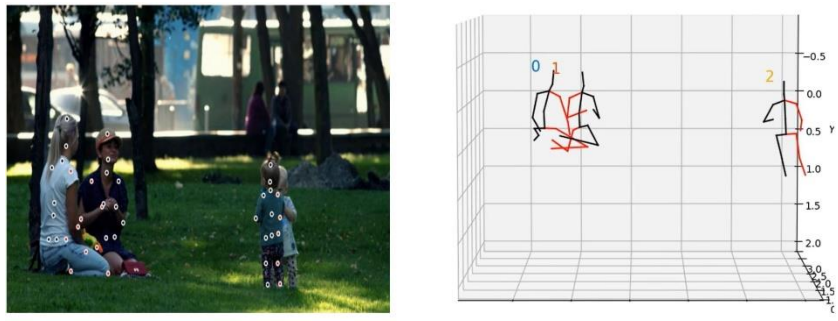

Fig 7 Challenging examples for POTR-3D

Chapter 7. Summary

In this paper, we present POTR-3D, a Transformer-based seq2seq 2D-to-3D approach for multi-person 3D pose estimation from monocular video. Introducing an additional frame-wise attention, we successfully extend the MixSTE [60] architecture from single-person to multi-person problem and empirically verify that this approach is indeed more robust on occlusion.

Moreover, we propose four types of data augmentation strategy to generate unlimited number of 2D-3D pair dataset, directly resolving the data scarcity issue innate in the 3D multi-person pose estimation problem.

The effectiveness of our approach is verified not just by achieving state-of-the-art performance on public benchmarks, MuPoTS-3D and CMU-Panoptic, but also by demonstrating accurate and consistent results on various in-the-wild videos.

On the other hand, there are some limitations on our approach. First, as POTR-3D does not receive any image or video information, it cannot estimate the exact size of person. It might struggle to distinguish a child nearby camera from an adult far from it. Second, we constrain the number of people to be consistent throughout this study for convenience. This should be relaxed for more practical use. Computationally, self-attention cost may dramatically increase as

the number of people in the video gets larger. Further approximation to reduce the self-attention cost may be needed.

Bibliography

[1] Karteek Alahari, Guillaume Seguin, Josef Sivic, and Ivan Laptev.Pose estimation and segmentation of people in 3d movies. In ICCV, 2013. 1

[2] Cristian Sminchisescu Andrei Zanfir, Elisabeta Marinoiu. Monocular 3D pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In CVPR, 2018. 2, 6

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. ViVit: A video vision transformer. In ICCV, 2021. 1

[4] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. Multi-person 3D pose estimation and tracking in sports. In CVPR Workshops, 2019. 1

[5] Simon Bultmann and Sven Behnke. Real-time multi-view 3D human pose estimation using semantic feedback to smart edge sensors. arXiv:2106.14729, 2021. 2

[6] 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 ICCV, 2019. 2

[7] Yu Cheng, Bo Wang, Bo Yang, and Robby T. Tan. Monocular 3D multi-person pose estimation by integrating top-down and bottom-up networks. In CVPR, 2021. 6, 7

[8] Yu Cheng, Bo Yang, Bo Wang, and Robby T Tan. 3D human pose estimation using spatio-temporal networks with explicit occlusion training. In AAAI, 2020. 2

[9] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In ICCV, 2019. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa

Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929, 2020. 2

[11] Miniar Ben Gamra and Moulay A Akhloufi. A review of deep learning techniques for 2D and 3D human pose estimation. Image and Vision Computing, 114:104282, 2021. 2

[12] Mohsen Gholami, Bastian Wandt, Helge Rhodin, RababWard, and Z Jane Wang. AdaptPose: Cross-dataset adaptation for 3D human pose estimation by learnable motion generation. In CVPR, 2022. 2

[13] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. PoseAug: A differentiable pose augmentation framework for 3D human pose estimation. In CVPR, 2021. 2

[14] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In CVPR, 2020. 2

[15] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997. 2

[16] Michael Hofmann and Dariu M Gavrila. Multi-view 3D human pose estimation combining single-frame recovery, temporal integration and model adaptation. In CVPR, 2009. 2

[17] Michael Hofmann and Dariu M Gavrila. Multi-view 3D human pose estimation in complex environment. International journal of computer vision, 96(1):103–124, 2012. 2

[18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human 3.6M: 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, 2013. 2

[19] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In ICCV, 2019. 2

[20] Lei Jin, Chenyang Xu, Xiaojuan Wang, Yabo Xiao, Yandong Guo, Xuecheng Nie, and Jian Zhao. Single-stage is enough: Multi-person absolute 3D pose estimation. In CVPR, 2022. 7, 8

[21] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain

Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In ICCV, 2015. 2, 6

[22] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. IEEE multimedia, 4(1):34–47, 1997. 2

[23] Shian-Ru Ke, LiangJia Zhu, Jenq-Neng Hwang, Hung-I Pai, Kung-Ming Lan, and Chih-Pin Liao. Real-time 3D human pose estimation from monocular view with applications to event detection and video gaming. In Proc. of the IEEE International Conference on Advanced Video and Signal Based Surveillance, 2010. 1

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014. 6

[25] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2016. 2

[26] Jogendra Nath Kundu, Ambareesh Revanur, Govind Vitthal Waghmare, Rahul Mysore Venkatesh, and R Venkatesh Babu. Unsupervised cross-modal alignment for multi-person 3D pose estimation. In ECCV, 2020. 2

[27] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3D human pose estimation. IEEE Transactions on Multimedia, 2022. 2

[28] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13147–13156, 2022. 2

[29] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3D pose sequence machines. In CVPR, 2017. 2

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll´ar, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In ECCV,

2014. i

[31] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In ECCV, 2020. 2

[32] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vil-BERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. NeurIPS, 32, 2019. 19

[33] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In ICCV, 2017. 2

[34] 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. In 3D Vision (3DV), 2017 Fifth International Conference on. IEEE, 2017. 2, 6

[35] 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. In 3D Vision (3DV), 2018 Sixth International Conference on. IEEE, 2018. 6

[36] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3D multi-person pose estimation from a single rgb image. In ICCV, 2019. 2, 6, 7, 8

[37] Francesc 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, pages 2823–2832, 2017. 2

[38] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016. 2

[39] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In ICCV, 2019. 2, 6

[40] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In CVPR, 2018. 2

[41] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and

Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In CVPR, 2019. 1, 2, 4, 6

[42] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3D human pose estimation. In ICCV, 2019. 2

[43] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. In CVPR, 2020. 2

[44] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In CVPR, 2017. 2

[45] Jiajun Su, Chunyu Wang, Xiaoxuan Ma, Wenjun Zeng, and Yizhou Wang. VirtualPose: Learning generalizable 3d human pose models from virtual data. arXiv:2207.09949, 2022. 6, 7, 8

[46] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In ICCV, 2019. 1

[47] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In CVPR, 2019. 3, 6, 7, i

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017. 2

[49] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 43(10):3349–3364, 2020. 2

[50] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3D human pose estimation: A review. Computer Vision and Image Understanding, 210:103225, 2021. 2

[51] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video

conferencing, 2020. 1

[52] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In CVPR, 2016. 2

[53] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In WACV, 2018. 6

[54] Qingqiang Wu, Guanghua Xu, Sicong Zhang, Yu Li, and Fan Wei. Human 3D pose estimation in a lying position by rgb-d images for medical diagnosis and rehabilitation. In Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020. 1

[55] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In ECCV, 2018. 2

[56] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In ICCV, 2021. 2

[57] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3D sensing of multiple people in natural images. In NeurIPS, 2018. 2, 6, 8

[58] Bowen Zhang, Hexiang Hu, Joonseok Lee, Ming Zhao, Sheide Chammas, Vihan Jain, Eugene Ie, and Fei Sha. A hierarchical multi-modal encoder for moment localization in video corpus. arXiv:2011.09046, 2020. 1

[59] Jianfeng Zhang, Yujun Cai, Shuicheng Yan, Jiashi Feng, et al. Direct multi-view multi-person 3d pose estimation. In NeurIPS, 2021. 2

[60] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video. arXiv preprint arXiv:2203.00859, 2022. 1, 2, 3, 8

[61] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In ECCV, 2022. 3, 6, i

[62] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human

pose regression. In CVPR, 2019. 2

[63] Jianan Zhen, Qi Fang, Jiaming Sun, Wentao Liu, Wei Jiang, Hujun Bao, and Xiaowei Zhou. Smap: Single-shot multi-person absolute 3d pose estimation. In European Conference on Computer Vision, pages 550–566. Springer, 2020. 6, 7, 8

[64] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. In ICCV, 2021.2

Abstract

컴퓨터 비전에 기반한 3차원 자세 추정(3D Pose Estimation)은 매우 다양한 분야에 응용될 수 있기 때문에 큰 가치가 있다. 최근, 트랜스포머(Transformer) 모델 기반의 시퀀스-시퀀스(Sequence-to-sequence) 모델인 MixSTE [60] 은 단일 객체(사람) 3차원 자세 추정에서 2차원 자세로부터의 3차원 자세 추정(2D-to-3D Lifting)의 방법을 활용하여 성공적인 결과를 거둔 바 있다. 본 연구는 이의 확장으로써 다중 객체 3차원 자세 문제를 다루며, 기존 연구와 비교해 등장하는 객체간 정보의 상호 참조(Inter-Personal Attention) 모듈을 새로이 추가하였다. 모델 구조에 기반하여 상호 인접 프레임 정보를 자연스럽게 참조함으로써, 본 연구에서 고안한 모델은 상호 가려짐 현상에 강인한 성능을 보였다. 하지만, 다중 객체 3차원 자세 추정은 데이터 부족 현상이라는 고질적인 문제를 지닌다. 본 연구의 방법론은 픽셀 수준의 디테일에서 벗어나, 2차원 자세와 3차원 자세 간의 관계를 다루기에, 주어진 데이터와 카메라 파라미터에 기반하여 데이터를 사실상 무제한적으로 증강할 수 있다는 강점을 지닌다. 본 분야에서 성능 측정 및 비교를 위한 대표적인 실험용 데이터셋에서 성능을 측정한 결과, 본 연구에서 고안한 모델은 정확도 뿐만 아니라 출력 결과의 부드러움 두 측면에서 모두 여타 기존 모델과 비교해 가장 훌륭한 성능을 보였다. 나아가, 테스트용 데이터셋 뿐만 아니라 다양한 시중 비디오에서도 훌륭한 성능을 보임으로써 연구의 상업적 가치 또한 입증하였다.