MoCaNet: Motion Retargeting in-the-wild via Canonicalization Networks

Wentao Zhu,[*1,2] **Zhuoqian Yang,**[*3] **Ziang Di,**[4] **Wayne Wu,**[2,3†] **Yizhou Wang,**[1] **Chen Change Loy**[5]

[1] School of Computer Science, Peking University [2] Shanghai AI Laboratory [3] SenseTime Research
[4] Southeast University [5] S-Lab, Nanyang Technological University
wtzhu@pku.edu.cn, yangzhuoqian@sensetime.com, 213180794@seu.edu.cn,
wuwenyan0503@gmail.com, yizhou.wang@pku.edu.cn, ccloy@ntu.edu.sg

Abstract

We present a novel framework that brings the 3D motion retargeting task from controlled environments to in-the-wild scenarios. In particular, our method is capable of retargeting body motion from a character in a 2D monocular video to a 3D character without using any motion capture system or 3D reconstruction procedure. It is designed to leverage massive online videos for unsupervised training, requiring neither 3D annotations nor motion-body pairing information. The proposed method is built upon two novel canonicalization operations, structure canonicalization and view canonicalization. Trained with the canonicalization operations and the derived regularizations, our method learns to factorize a skeleton sequence into three independent semantic subspaces, *i.e.*, motion, structure, and view angle. The disentangled representation enables motion retargeting from 2D to 3D with high precision. Our method achieves superior performance on motion transfer benchmarks with large body variations and challenging actions. Notably, the canonicalized skeleton sequence could serve as a disentangled and interpretable representation of human motion that benefits action analysis and motion retrieval.

Introduction

3D motion retargeting aims at transferring one character's motion to a virtual 3D avatar. It is an important and challenging task in computer vision and computer graphics with a wide spectrum of applications in human-computer interaction (Hoshyari et al. 2019; Kim and Lee 2020) and augmented reality (Kang et al. 2019; Kim et al. 2016). Traditional approaches rely on motion capture (MoCap) systems (Ott, Lee, and Nakamura 2008; Koenemann, Burget, and Bennewitz 2014), which are capable of capturing precise 3D motion but are limited to sophisticated environments only available to film and game studios. This kind of setup is inaccessible for everyday users with in-the-wild usage scenarios, *e.g.*, on-device augmented reality.

The challenge of in-the-wild motion retargeting lies in that videos recorded on mobile devices or downloaded from the Internet only provide 2D visual information (usually noisy), from which 3D motion needs to be inferred. The common practice is to estimate 3D pose first, then perform 3D-to-3D retargeting through Inverse Kinematics (IK). Nevertheless, the results of current 3D pose estimation (Ci et al. 2019; Pavllo et al. 2019; Moon, Chang, and Lee 2019; Yang et al. 2018) and model-based mesh reconstruction (Liu et al. 2019; Kocabas, Athanasiou, and Black 2020; Choi, Moon, and Lee 2020) methods deteriorate severely under occlusions, large body variations and uncommon actions from unconstrained real-world videos, as discussed in (Dong et al. 2020; Chen et al. 2019b; Duan et al. 2021). The errors originated from 3D pose estimation can easily propagate to the IK stage and degrade the quality of retargeted motion.

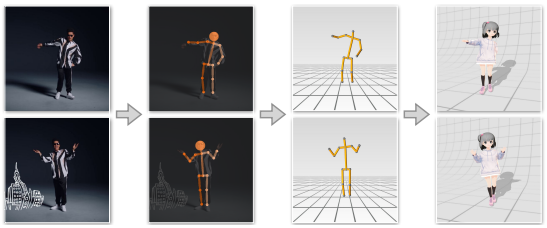


Figure 1: **3D Motion Retargeting in the Wild.** Motion is extracted from the video clip and retargeted to the virtual avatar. Our method extracts 2D skeleton sequences rather than 3D sequences from in-the-wild videos.

In this work, we argue that the conventional way of performing IK after 3D estimation is not sufficiently robust and reliable for in-the-wild motion retargeting. To this end, we propose an end-to-end learnable model, *Canonicalization Networks*, which allows us to bypass the error-prone direct 3D pose estimation and use an off-the-shelf 2D pose estimator to extract geometrical information that is more robust and reliable. Our method then takes the 2D skeleton sequences and generates 3D retargeted skeleton sequences, as shown in Fig. 1. However, retargeting motion from 2D skeleton sequences is not a trivial task. Arbitrarily captured real-world videos contain large structural variations and diverse camera views. The same action, *e.g.*, standing up, may appear dramatically different in 2D under different views given different body structures. It is therefore meaningful and challenging to extract the unmixed motion in a high-level semantic modality.

We tackle the challenge through the novel notion of

[*]These authors contributed equally.
[†]Corresponding author.

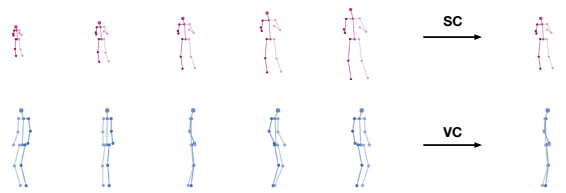


Figure 2: **Canonicalization Operations.** The first row shows the idea of *Structure Canonicalization*, and the second row shows the idea of *View Canonicalization*. Each skeleton represents an individual video clip.

canonicalization and retargeting. Canonicalization, by definition, aims at eliminating variations in a specific domain. In our context, we explore the canonicalization of structure and view angle from 2D skeleton sequences while keeping the motion unchanged. Specifically, we design two parallel canonicalization operations, structure canonicalization and view canonicalization, as shown in Fig. 2. Structure canonicalization yields skeleton sequences with a uniform body structure, while the motion and other features are preserved. Similarly, view canonicalization provides skeleton sequences with the same view angle, casting different sequences to a uniform view. The idea behind is that a 2D skeleton sequence can be formulated as a product of three independent latent variables, namely, motion, structure, and view. By learning the canonicalization operations, the model should be able to disentangle the three meaningful factors from 2D skeleton sequences. It then allows us to freely recombine motion with arbitrary structures and view angles to generate the desired 3D output, as shown in Fig. 3.

Based on these canonicalization operations, we formulate a set of novel self-supervised canonicalization losses, which randomly perturb a targeted factor (structure or view angle subspaces) and apply canonicalization in that space for both the original sequence and the manipulated sequence. Hence, the model can be trained on an extensive collection of human pose sequences extracted from Internet videos without any annotation, which notably increases the robustness and generalization of the model. Thanks to canonicalization and unsupervised training, the network is exposed to a wide range of variations. Such exposure and the requirement of restoring them to a specific canonical form (structure and view) allows the network to learn focusing on the inherent motion and omitting noisy information from the input sequence. This contributes to learning well-disentangled representations for precise motion retargeting.

The contributions of this work are three-fold: 1) We propose an end-to-end learnable model, *Canonicalization Networks* to tackle the challenging in-the-wild 2D-to-3D motion retargeting problem. 2) We design an unsupervised learning approach based on the novel canonicalization operations, requiring neither paired data nor 3D annotations. Our approach circumvents the errors in monocular 3D estimation and benefits from large-scale web data for unsupervised training, which leads to improved performance demonstrated in our experiments. 3) Canonicalized skeleton sequences could serve as an interpretable representation of human motion, paving the way for related research and applications.

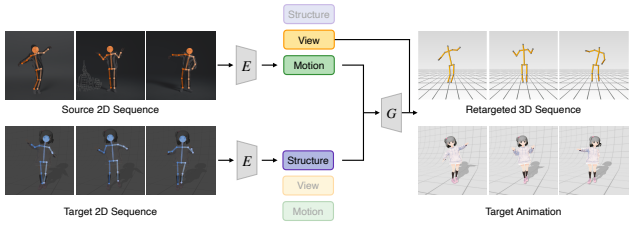


Figure 3: **Motion Retargeting Inference Pipeline.** The encoder, E, decomposes the input skeleton sequence into three latent codes. The decoder, G, takes the motion code from the source character and the structure code from the target character, then cast the decoded 3D skeleton sequence to the source view angle. This yields the retargeted 3D sequence, which could be used to animate the target character.

Related Work

Motion Retargeting. Motion retargeting has been extensively studied in the area of computer vision and graphics. Classical motion retargeting methods (Hodgins and Pollard 1997; Gleicher 1998; Tak and Ko 2005; Lee and Shin 1999) mainly rely on simplified assumptions and hand-crafted kinematic constraints. In recent years, deep-learning-based approaches show promising results on obtaining human pose and action, which not only increases the availability of motion data, but also inspires motion retargeting research with deep neural networks. Villegas et al. proposed to capture high-level motion with Forward Kinematics (FK) layers in recurrent neural networks. Lim, Chang, and Choi further optimized the retargeting precision by disentangling pose and movement. Skeleton-Aware Networks (Aberman et al. 2020) could automatically adapt to different skeleton topologies. Nevertheless, the above methods all require high-precision 3D motion (*e.g.*, quaternions), which takes expensive motion capture systems or complex optimization in practice. Otherwise, estimating 3D motion from monocular RGB video could be error-prone (Dong et al. 2020; Chen et al. 2019b; Duan et al. 2021).

Several works have also explored to retarget motion from 2D inputs that are more accessible to everyday use cases. Aberman et al. proposed a two-branch framework with part confidence map as 2D pose representations. Chan et al. designed a global pose normalization to handle different body structures. Aberman et al. proposed a supervised learning approach to retarget motion in 2D. It requires the exact same motion performed by different characters at different view angles as training data. TransMoMo (Yang et al. 2020) explored unsupervised motion retargeting via invariance properties. These methods all generate 2D label maps for further rendering via image-to-image translation methods. Therefore, they can not be readily applied to driving 3D characters. They only consider accuracy for 2D joint positions and assume static view angle. Meanwhile, our canonicalization-based approach enables applying regularization on 3D poses directly and training with time-varying explicit 3D view.

In our framework, 2D poses are used as input and our method produces 3D skeletons as output. Therefore, our

approach can exploit user-friendly monocular RGB videos with the help of robust 2D pose estimation algorithms (Cao et al. 2018; Bulat et al. 2020; Artacho and Savakis 2020). In addition, the 3D skeleton output allows combination with state-of-the-art neural rendering (Wang et al. 2018, 2019) and avatar creation (Saito et al. 2020; Deng et al. 2020; Alldieck et al. 2018; Weng, Curless, and Kemelmacher-Shlizerman 2019) techniques. Our flexible approach thus provides a viable alternative to the traditional 3D reconstruction+IK pipeline.

Representation Disentanglement. Disentangling latent variables from observations has been a fundamental topic in machine learning. Previous approaches (Kingma et al. 2014; Mathieu et al. 2016) use labeled data to separate class-dependent and class-independent features. A vast literature focuses on unsupervised learning of disentangled representations with generative models (Chen et al. 2016; Denton and Birodkar 2017; Villegas et al. 2017; Tulyakov et al. 2018; Higgins et al. 2017; Lee et al. 2018; Huang et al. 2018). A recent line of work explored view-angle-agnostic motion representations aimed at improving motion recognition performance (Liu et al. 2021; Nie, Liu, and Liu 2020; Zhao et al. 2021). In this work, we achieve disentanglement of motion, body-structure and view-angle via unsupervised canonicalization.

3D Canonicalization. Several different methods have been proposed for automatically computing the canonical form from a 3D pose or mesh. (Pickup et al. 2015b,a, 2018, 2016) propose to learn pose-neutral shape of any non-rigid meshes for 3D shape retrieval. (Shamai, Zibulevsky, and Kimmel 2015) further improves the method with multidimensional scaling. Lian, Godil, and Xiao present a feature-preserved canonical form that directly deforms the 3D models. Body reconstruction methods (Saito et al. 2020; Wang, Geiger, and Tang 2021) sometimes involve the canonical pose space for handling pose-dependent deformations. In those papers, canonicalization refers to *pose canonicalization*, that is, "transforming to a standard pose". In this work, we explore the canonicalization of different factors, namely structure and view angle, while keeping the motion unchanged. C3DPO (Novotny et al. 2019) designed a rotation-invariant canonical shape for structure reconstruction. CanonPose (Wandt et al. 2021) learns a canonical camera view and respective camera rotations, but it relies on multiple synchronized cameras for training. We design canonicalization operations for both structure and view angle, which are applied to the entire action sequence rather than a single pose. Based on the designed operations, our model can be trained without data pairing.

Methodology

Overview

Our solution to in-the-wild 3D motion retargeting uses 2D skeleton sequences for training and inference. Specifically, we extract pose sequences from unannotated web videos with an off-the-shelf human pose estimator (Alp Güler, Neverova, and Kokkinos 2018; Girshick et al. 2018). We devise a 2D-to-3D autoencoder that factorizes the input 2D human pose sequence into three independent spaces, namely character-agnostic 3D motion, body structure of the character and view angle. Notably, we formulate the view angle to be explicit and dynamic, which allows for direct manipulation. 3D motion retargeting is then achieved through decoding a recombination of motion and structure encoded from different sources.

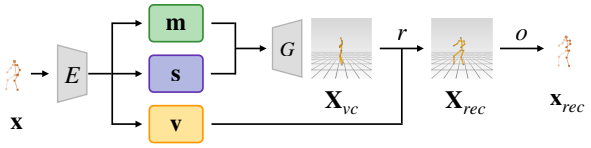


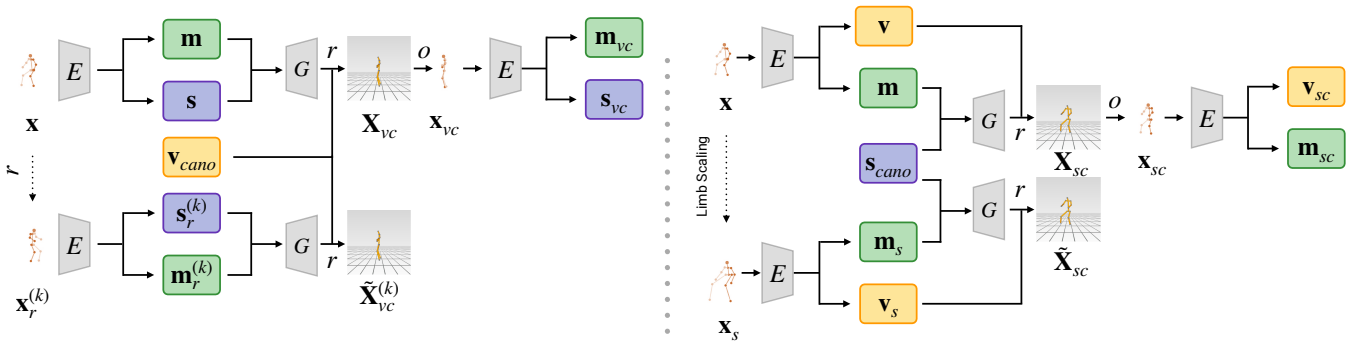
Figure 4: **2D-to-3D AutoEncoder Structure.** The encoder E encodes an input 2D skeleton sequence $\mathbf{x}$ as character-agnostic motion $\mathbf{m}$, structure $\mathbf{s}$ and view $\mathbf{v}$. The decoder G takes as input the concatenation of $\mathbf{m}$ and $\mathbf{s}$ and outputs a 3D skeleton sequence $\mathbf{X}_{vc}$ in canonical view. The reconstructed 3D sequence in original camera view is obtained by rotating $\mathbf{X}_{vc}$ with the encoded view angles $\mathbf{v}$, *i.e.*, $\mathbf{X}_{rec} = r(\mathbf{X}_{vc}, g(\mathbf{v}))$. The function $g(\cdot)$ converts the view $\mathbf{v}$ to rotation matrices. The reconstructed 2D sequence $\mathbf{x}_{rec}$ is obtained after an orthographic projection, *i.e.*, $\mathbf{x}_{rec} = o(\mathbf{X}_{rec})$.

We design a training scheme that simultaneously achieves (i) 3D geometry inference and (ii) disentangled motion representation learning. The two goals are tightly coupled and mutually beneficial. The first goal is addressed via a 2D-to-3D AutoEncoder trained with adversarial learning (Chen et al. 2019a). The second challenge is addressed via the proposed self-supervised canonicalization.

2D-to-3D AutoEncoder

As shown in Fig. 4, the proposed 2D-to-3D AutoEncoder is a modular encoder-decoder network, where the encoding part decomposes an input 2D skeleton sequence $\mathbf{x} \in \mathbb{R}^{2N \times T}$ into three spaces: motion, structure and view angle, while the decoding part takes any combination of motion and structure and decodes a 3D skeleton sequence. Here, N denotes the number of body joints and T denotes the sequence length. The decoded 3D skeleton sequence should reproduce the input when projected back to 2D while remain plausible when it is viewed at different angles. This serves as the foundation of the representation learning framework and allows us to manipulate the data for canonicalization training.

Encoders. The encoded representation of the three spaces are defined as follows: Motion is represented as a temporal sequence of latent vectors $E_m(\mathbf{x}) = \mathbf{m} \in \mathbb{R}^{M \times C_m}$ where M is the encoded length and C_m is the number of latent channels. The motion encoder uses several 1-D temporal convolutional layers to extract this information, yielding the encoded length $M = T/8$. The structure encoder has a similar network structure $E_s(\mathbf{x}) = \mathbf{s} \in \mathbb{R}^{M \times C_s} = [\mathbf{s}^{(1)}, \mathbf{s}^{(1)}, ..., \mathbf{s}^{(M)}]$. This can be interpreted as performing multiple structure estimations in sliding windows. Since the structure does not change with time, we use a temporal max pooling to aggregate information from these estimations to get the final structure: $\bar{E}_s(\mathbf{x}) = \bar{\mathbf{s}} = \text{maxpool}(\mathbf{s}), \bar{\mathbf{s}} \in \mathbb{R}^{C_s}$. We assume view angle to be dynamic, therefore it is encoded



View Canonicalization **Structure Canonicalization**

Figure 5: **Canonicalization.** (a) View canonicalization $\phi_{vc}(\cdot)$ is to restore canonical view angle $\mathbf{v}_{cano}$ for a skeleton sequence. During training, view canonicalization is applied to both the input 2D sequence $\mathbf{x}$ and its randomly rotated versions $\mathbf{x}_r^{(k)}$, *i.e.*, $\mathbf{X}_{vc} = \phi_{vc}(\mathbf{x}), \tilde{\mathbf{X}}_{vc}^{(k)} = \phi_{vc}(\mathbf{x}_r^{(k)})$. (b) Structure canonicalization $\phi_{sc}(\cdot)$ is to store standard body structure $\mathbf{s}_{cano}$ for a skeleton sequence. During training, structure canonicalization is applied to both the input 2D sequence $\mathbf{x}$ and its randomly limb-scaled version $\mathbf{x}_s$, *i.e.*, $\mathbf{X}_{sc} = \phi_{sc}(\mathbf{x}), \tilde{\mathbf{X}}_{sc} = \phi_{sc}(\mathbf{x}_s)$.

as a temporal sequence of explicit 3D rotations. Instead of the commonly used Euler angles or quaternions, we choose to use a continuous 6D representation, which is better suited for deep learning regression (Zhou et al. 2019). Specifically, $E_v(\mathbf{x}) = \mathbf{v} \in \mathbb{R}^{T\times 6}$, *i.e.*, $\mathbf{v}$ is a sequence of 6D rotation vectors, each rotation vector $\mathbf{v}_i$ is responsible for rotating one frame. $\mathbf{v}$ can be uniquely translated to a sequence of rotation matrices $\mathbf{\Omega} \in \mathbb{R}^{T\times 3\times 3} = [\boldsymbol{\omega}^{(1)}, \boldsymbol{\omega}^{(2)}, ..., \boldsymbol{\omega}^{(T)}], \boldsymbol{\omega}^{(i)} \in SO3$ using a function: $g(\mathbf{v}) = \mathbf{\Omega}$.

Decoder. As shown in Fig. 4, the decoder takes the concatenation of the motion and structure as input to decode a 3D skeleton sequence $G(\mathbf{m}, \mathbf{s}) = \mathbf{X}_{vc} \in \mathbb{R}^{3N\times T}$. The decoded sequence is assumed to be in the canonical view angle. To reproduce the input 2D sequence, the output 3D skeleton sequence $\mathbf{X}_{vc}$ is rotated using the encoded view angles $\mathbf{v}$ and then projected to 2D, *i.e.*, $\mathbf{X}_{rec} = r(\mathbf{X}_{vc}, g(\mathbf{v}))$, $\mathbf{x}_{rec} = o(\mathbf{X}_{rec})$. The function $r(\mathbf{X}, \mathbf{\Omega})$ is one which rotates a 3D skeleton sequence $\mathbf{X}$ with a sequence of SO3 rotation matrices $\mathbf{\Omega}$. $o(\mathbf{X})$ stands for orthographic projection.

3D Self-Supervision. Since 3D ground-truth is not available, the ill-posed 2D-to-3D lifting is achieved via supervision in 2D space. Specifically, we use (i) an $L1$ reconstruction loss to ensure that the reprojected 2D skeleton sequence reproduces the input and (ii) an adversarial loss to ensure the reconstructed $3D$ skeleton, when viewed at random angles, still lies within the distribution of real $2D$ skeleton sequences. Specifically, the $L1$ reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}} = |\mathbf{x} - \mathbf{x}_{rec}|, \tag{1}$$

where $\mathbf{x}$ is the input 2D sequence and $\mathbf{x}_{rec}$ is the sequence reconstructed by the autoencoder. To calculate the adversarial loss, we rotate the lifted 3D sequence with some random view angles. We generate K versions of rotated sequences $\mathbf{X}_r^{(k)} = r(\mathbf{X}_{rec}, \mathbf{\Omega}_k), \mathbf{x}_r^{(k)} = o(\mathbf{X}_r^{(k)})$. Note that the rotation matrices are time-varying, and in practice we keep them temporally smooth by interpolation. A discriminator D is used for the adversarial training. The discriminator tries to distinguish between real and reprojected 2D sequences while the AutoEncoder tries to make its 3D output indistinguishable regardless of viewing angle. This process eliminates the ambiguity in the 2D-to-3D mapping by penalizing unusual structures in the reprojected views (Chen et al. 2019a). The adversarial loss is defined as

$$\mathcal{L}_{\text{adv}} = \sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}[\log D(\mathbf{x}) + \log(1 - D(\mathbf{x}_r^{(k)})] \tag{2}$$

Canonicalization

Canonicalization aims at restoring canonicality in one of the spaces while leaving the other two unchanged, a process that enforces the independence of one space against the other two. We propose view canonicalization (restore canonical view for any 3D skeleton sequence) and structure canonicalization (restore canonical body structure for any 3D skeleton sequence). We randomly manipulate the 3D sequence in only one of the three spaces and apply canonicalization in that space for both the original sequence and the manipulated sequence. This process gives us several variants that can be used to derive the supervision signals.

View Canonicalization. View canonicalization $\phi_{vc}(\mathbf{x})$ restores canonical view for any skeleton sequence, *i.e.*, performing view canonicalization on figures with arbitrary orientation should yield figures facing the same direction, while leaving the motion and body structure unchanged. This is achieved by reconstructing the 3D sequence with the decoder and then rotating it to canonical view, *i.e.*, $\phi_{vc}(\mathbf{x}) = r(G(E_m(\mathbf{x}), E_s(\mathbf{x})), g(\mathbf{v}_{cano}))$. The canonical view $\mathbf{v}_{cano}$ is defined as corresponding to the identity matrix, *i.e.*, no rotation. Thus, the output of the decoder is assumed to be in a canonical view, see Fig. 5 (a) for an example.

Recall in the previous section that we generated 2D reprojections $\mathbf{x}_r^{(k)}$ of the input sequence $\mathbf{x}$. For training, we perform view canonicalization on both the input and the reprojected versions. Since $\mathbf{x}$ and $\mathbf{x}_r^{(k)}$ contain the same motion

and structure and the only variable is the view angle, the results of view canonicalization should be identical. From this property we derive the view canonicalization loss in skeleton space:

$$\mathcal{L}_X^{\text{vc}} = \sum_{k=1}^{K} \left| \mathbf{X}_{vc} - \tilde{\mathbf{X}}_{vc}^{(k)} \right|, \tag{3}$$

where $\mathbf{X}_{vc} = \phi_{vc}(\mathbf{x})$ and $\tilde{\mathbf{X}}_{vc}^{(k)} = \phi_{vc}(\mathbf{x}_r^{(k)})$. Also, since rotation and view canonicalization does not modify information in the motion and structure space, the re-encoded motion and structure should stay the same. These properties give us the view canonicalization loss in feature space.

$$\mathcal{L}_m^{\text{vc}} = |\mathbf{m} - \mathbf{m}_{vc}| + \sum_{k=1}^{K} \left| \mathbf{m} - \mathbf{m}_r^{(k)} \right|, \tag{4}$$

$$\mathcal{L}_s^{\text{vc}} = |\mathbf{s} - \mathbf{s}_{vc}| + \sum_{k=1}^{K} \left| \mathbf{s} - \mathbf{s}_r^{(k)} \right|, \tag{5}$$

where $\mathbf{m}$ and $\mathbf{s}$ are the motion and structure encoded from the input sequence $\mathbf{x}$. Together, the *view canonicalization loss* is defined as

$$\mathcal{L}_{vc} = \lambda_X^{vc}\mathcal{L}_X^{\text{vc}} + \lambda_m^{vc}\mathcal{L}_m^{\text{vc}} + \lambda_s^{vc}\mathcal{L}_s^{\text{vc}}. \tag{6}$$

Structure Canonicalization. Structure canonicalization restores a standard body structure for any skeleton sequence while leaving the motion and view angle unchanged. During training, we estimate a centroid of the structure space and define it as the canonical structure $\mathbf{s}_{cano}$. Structure canonicalization is defined as $\phi_{sc}(\mathbf{x}) = r\left(G(E_m(\mathbf{x}), \mathbf{s}_{cano}), E_v(\mathbf{x})\right)$. The pipeline for structure canonicalization training is shown in Fig. 5 (b).

To achieve self-supervision, a technique called limb-scaling (Yang et al. 2020), *i.e.*, randomly shortening or extending the length of the limbs in the input 2D sequence $\mathbf{x}$, is used to obtain a sequence $\mathbf{x}_s$, which has the same motion and view angle but a different body structure. We then perform structure canonicalization on both sequences, *i.e.*, $\mathbf{X}_{sc} = \phi_{sc}(\mathbf{x})$, $\tilde{\mathbf{X}}_{sc} = \phi_{sc}(\mathbf{x}_s)$. By definition, this pair of canonicalization results should be the same, from which we derive the structure canonicalization loss in the skeleton space:

$$\mathcal{L}_X^{\text{sc}} = \left| \mathbf{X}_{sc} - \tilde{\mathbf{X}}_{sc} \right|. \tag{7}$$

Since limb-scaling and structure canonicalization does not modify information in the other two spaces, the re-encoded motion and view angle should stay the same. These properties give us the structure canonicalization loss in feature space.

$$\mathcal{L}_m^{\text{sc}} = |\mathbf{m} - \mathbf{m}_{sc}| + |\mathbf{m} - \mathbf{m}_s|, \tag{8}$$

$$\mathcal{L}_v^{\text{sc}} = |\mathbf{v} - \mathbf{v}_{sc}| + |\mathbf{v} - \mathbf{v}_s|. \tag{9}$$

Together, the *structure canonicalization loss* is defined as

$$\mathcal{L}_{sc} = \lambda_X^{sc}\mathcal{L}_X^{\text{sc}} + \lambda_m^{sc}\mathcal{L}_m^{\text{sc}} + \lambda_v^{sc}\mathcal{L}_v^{\text{sc}}. \tag{10}$$

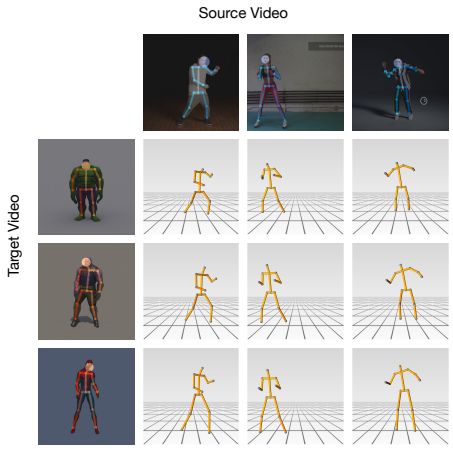


Figure 6: **3D Motion Retargeting in-the-Wild.** The uppermost row gives the source videos, and the leftmost column gives the target videos. For each source-target combination, the network transfers the motion from source to target and produces 3D results.

Total Loss

The total training loss is given as

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \mathcal{L}_{vc} + \mathcal{L}_{sc}. \tag{11}$$

We back-propagate $\mathcal{L}$ and train the whole network in an end-to-end fashion.

Experiments

In this section, we first evaluate and compare the motion retargeting performance on both in-the-wild and synthetic data. Then, we investigate the effects of canonicalization on the learned representations. We also conduct an ablation study to examine the effectiveness of each module. We include the details of the neural network and the datasets in the appendix due to space limit.

Motion Retargeting

Figure 6 shows the motion retargeting results of our method. Each output skeleton is conditioned on the motion and view code from the source character and the structure code from the target character. The results demonstrate that our method could transfer the motion naturally and realistically to the target body. In addition, we compare our method with the state-of-the-art methods. To implement the conventional 3D+IK pipeline, we use several SOTA 3D pose estimation and mesh reconstruction methods as detailed below. Two sets of results of our method are presented. The *Ours* model is trained on the synthetic Mixamo training set and the *Ours (wild)* model is trained on a web-crawled video dataset SoloDancer (Yang et al. 2020). Please refer to the supplementary materials for detailed setups and formal definitions of metrics.

Video-to-3D (In-the-Wild) Comparison. We first test under the in-the-wild setting where the source and target are given by dancing videos downloaded from the Internet, and the results are shown in Fig. 7. VideoPose3D (Pavllo

Table 1: **2D-to-3D (Synthetic) Motion Retargeting Comparison.** We calculate the 3D Mean Square Error (MSE $\times 10^{-2}$) and 3D Mean Per Joint Position Error (MPJPE $\times 10^{-1}$). Both MSE and MPJPE are root-relative and normalized by the target character's body height, following the practice of (Aberman et al. 2019b). The test characters are shown in Fig. 8.

Methods	3DGT+IK		Ours		Ours (wild)		TransMoMo		LCN+IK		Pose2Mesh+IK		VideoPose3D+IK	
Metrics	MSE	MPJPE	MSE	MPJPE	MSE	MPJPE	MSE	MPJPE	MSE	MPJPE	MSE	MPJPE	MSE	MPJPE
Andro. ↔ P. Hulk	0.049	0.325	0.799	1.221	0.792	1.215	1.246	1.563	1.986	1.925	2.822	2.361	3.198	2.469
Andro. ↔ S. Granny	0.042	0.294	0.948	1.320	0.945	1.320	1.668	1.830	1.973	1.881	2.504	2.159	5.309	3.173
Andro. ↔ TY	0.016	0.168	0.853	1.206	0.832	1.191	1.497	1.725	1.769	1.730	1.910	1.878	4.044	2.678
P. Hulk ↔ S. Granny	0.131	0.507	0.955	1.334	0.951	1.333	2.057	2.072	2.235	2.013	3.182	2.491	6.390	3.760
P. Hulk ↔ TY	0.064	0.371	0.874	1.230	0.853	1.220	1.837	1.965	2.135	1.942	2.544	2.235	5.471	3.429
S. Granny ↔ TY	0.024	0.209	0.927	1.266	0.912	1.256	2.166	2.147	2.141	1.886	2.313	2.073	7.542	4.077
OVERALL	0.056	0.316	0.891	1.261	0.878	1.253	1.750	1.888	2.046	1.899	2.552	2.204	5.329	3.272

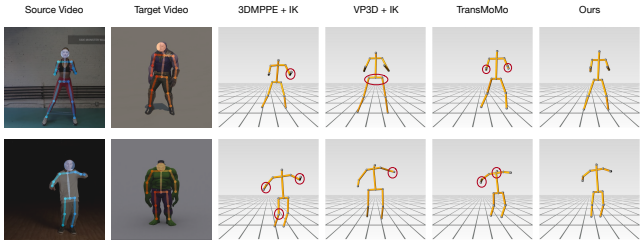


Figure 7: **Video-to-3D (In-the-Wild) Motion Retargeting Comparison.** The compared methods retarget the motion from source videos to target videos in the form of 3D skeleton sequences. Please zoom in for details, especially those circled with red that shows the erroneous results produced by existing methods.

Figure 8: **Mixamo Testset Characters.**

et al. 2019) takes a 2D skeleton sequence and lifts it to 3D with temporal convolutions. 3DMPPE (Moon, Chang, and Lee 2019) regresses 3D pose from the 2D image directly. In the traditional two-stage pipeline, errors in the 3D pose estimation stage are accumulated and enlarged in the IK stage. Therefore, these methods are highly reliant on the quality of 3D pose estimation, which tends to fail due to occlusions or short 2D projections. TransMoMo (Yang et al. 2020) produces an intermediate 3D result before projection. However, it is designed to generate 2D label maps for further rendering, so they only consider 2D supervision with an oversimplified static view assumption. Although its 2D projection results are plausible, multiple artifacts can be found in 3D, *e.g.*, wrong direction, temporally twisted moves.

2D-to-3D (Synthetic) Comparison. In order to quantitatively compare the motion retargeting performance with the previous methods, we compute the 2D-to-3D motion retargeting error on a synthetic animation dataset called Mixamo (Adobe 2021) because it has ground-truths available. The training set comprises 32 characters, and each character has 800 actions. During evaluation, we test the models on a held-out partition with 4 new characters (Fig. 8) and 64 new actions. Since 3DMPPE (Moon, Chang, and Lee 2019) is not applicable to the 2D pose input, we further add two recent SOTA methods LCN (Ci et al. 2020) and Pose2Mesh (Choi, Moon, and Lee 2020) for the conventional pipeline. We further compare a strong benchmark where ground-truth 3D motion is retargeted with IK. It resembles the practice of using high-precision MoCap systems and serves as a lower bound of the error. The results are shown in Table 1. Our approach outperforms previous methods by a considerable margin on all the test character pairs. The performance of *3DGT+IK* (the upper-bound baseline with 3D pose grouth-truths) proves that the IK step itself is almost error-free. Hence, errors are mainly produced at the 3D stage and amplified in the IK stage as for the conventional two-stage pipeline. It is also notable that Solo-Dancer is only 30% of Mixamo training set in terms of video length. The model trained on Solo-Dancer performs better, which demonstrates the strength of training on in-the-wild data with higher motion diversity.

Canonicalization and Disentanglement

By canonicalization training, the model learns to disentangle the three latent factors without external supervision. We first visualize the canonicalization results, which shows that the model learns to transform the skeleton sequences as intended. In addition, we show that the learned canonical form could serve as an effective representation of human motion.

Canonicalization Effects. As shown in Fig. 9, input skeleton sequences with different orientations are aligned to the same view angle after view canonicalization (Row 2), while the motion and structure information are not changed. The view canonicalization requires the network to infer the underlying 3D structure of the input motion sequence and automatically cast it to a specific view. Despite the large input body structure discrepancies, structure canonicalization (Row 3) generates a "standard character" performing the same motion at the exact view angle. The structure canonicalization requires the network to capture the average structure and learn a character-agnostic motion representation. As shown in the last row of Fig. 9, the 3D skeleton sequence after both structure and view canonicalization contains unmixed motion information, independent of body structure and view angle. Therefore, it can be used as a direct, disentangled, and interpretable motion representation. We conduct further experiments to show the practicability of using the dual-canonicalized skeleton sequence as a distilled mo-

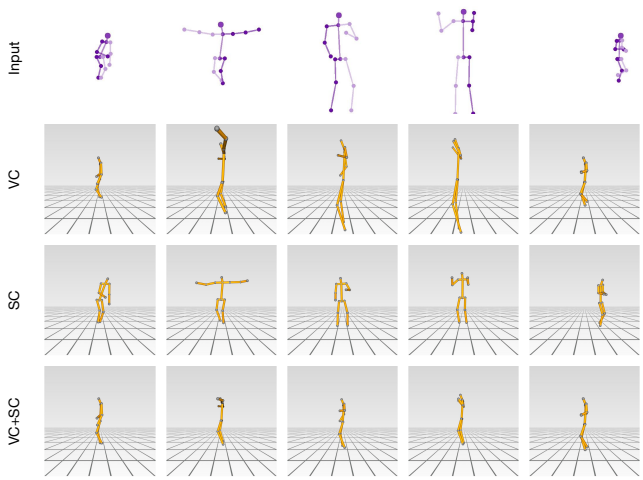


Figure 9: **Canonicalization Results.** The first row is the input 2D skeleton sequences, all from the Mixamo test set. The following rows show the 3D results after applying view canonicalization, structure canonicalization, and both.

Table 2: **Motion Clustering Evaluation.** For all the metrics, larger number means better clustering result.

	ARI	AMI	Homogeneity	Completeness	V-Measure
TransMoMo	0.241	0.620	0.657	0.756	0.703
Canonical	**0.347**	**0.707**	**0.750**	**0.808**	**0.778**

tion representation.

Motion Clustering. We apply a simple K-Means algorithm to cluster the Mixamo test set by motions. The dataset consists of 64 different motions performed by 4 characters and observed at 7 view angles. For K-Means, we set $K = 64$ and iterate 300 times. The clustering results are evaluated by ARI (Adjusted Rand index) (Hubert and Arabie 1985), AMI (Adjusted Mutual Information) (Gleicher 1998), Homogeneity, Completeness, and V-Measure (Rosenberg and Hirschberg 2007). We compare two motion representations: 1) Latent motion embedding learned by TransMoMo (Yang et al. 2020), and 2) Skeleton sequence after our canonicalization operations. Table 2 shows that our canonical motion representation consistently outperforms the previous work. It suggests that the proposed canonicalization operations help to learn better motion representation.

Motion Retrieval. The dual-canonicalized motion representation also enables the motion retrieval task, *i.e.*, retrieve videos that exhibit similar motions with the query video. We first extract pose from the Solo-Dancer dataset, then slice the pose sequences into clips of 64 frames, producing a motion library of 33678 clips. Then, for a query motion, we retrieve its nearest neighbors in the dual-canonicalized motion representation space. The results in Fig. 10 show that our method is able to retrieve videos with similar motion semantics even if they appear drastically different due to body structure and view angle variations. It is because canonicalization operations make the proposed representation independent of both structure and view angle. For video results of motion retrieval, please refer to the attached video.

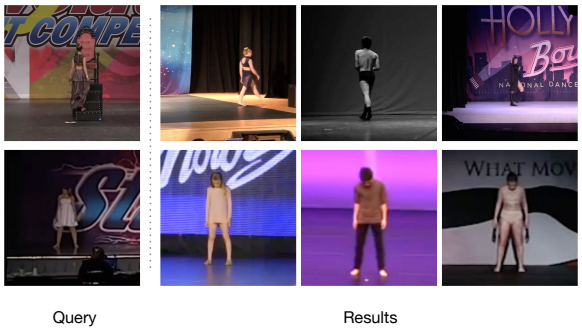


Figure 10: **Motion Retrieval Results.** The leftmost column shows a frame sampled from a query video depicting a short motion, and the three other columns show the top three retrieved videos.

Table 3: **Ablation Study.** *w/o* adv refers to the model with adversarial loss ablated, *w/o* VC refers to the model with view canonicalization loss ablated, *w/o* SC refers to the model with structure canonicalization loss ablated. T(3D) refers to TransMoMo re-implemented with 3D loss terms. T(3D+DV) refers to TransMoMo trained with 3D loss terms and dynamic view. The numbers are measured on the Mixamo dataset.

	w/o adv	*w/o* VC	*w/o* SC	T(3D)	T(3D+DV)	Ours (Full)
MSE $\times 10^{-2}$	1.579	1.240	1.599	1.652	1.688	**0.891**
MPJPE $\times 10^{-1}$	1.785	1.602	1.693	1.776	1.743	**1.261**

Ablation Study

We first ablate several loss terms to show the importance of the designed modules. Then, we provide a stepwise comparison with TransMoMo (Yang et al. 2020). In Table 3, the removal of any part of our full model results in non-trivial performance degradation. More specifically, we observe that the network trained without adversarial loss tends to generate poses with a wrong depth, and the network trained without canonicalization loss fails to keep the desired motion. Although directly applying dynamic view and 3D loss terms to TransMoMo (Yang et al. 2020) slightly helps, the models still exhibit a considerable performance gap compared to the proposed canonicalization-based approach. The models also fail to learn time-varying 3D view angle as expected. We argue that the proposed canonicalization operations not only help to learn disentangled motion better, but also ease the learning of dynamic view representations, which are more realistic yet more complex.

Conclusion

In this work, we present Canonicalization Networks for tackling in-the-wild 2D-to-3D motion retargeting. The proposed method robustly infers and transfers motion from 2D monocular videos to drive 3D humanoids, enabling motion retargeting in everyday use cases. We derive self-supervision from the canonicalization operations in factorized semantic spaces, from which the presented framework can be trained easily with web-crawled videos only given no annotations. To sum up, our 3D motion retargeting approach enables both training and inference in-the-wild. The experiment results show improvement of the proposed method over the existing methods, especially in terms of robustness and generalization.

Acknowledgement. This study is partially supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). We thank Xiaoxuan Ma, Shikai Li, Jiajun Su and Peizhuo Li for insightful discussion and kind support.

References

Aberman, K.; Li, P.; Lischinski, D.; Sorkine-Hornung, O.; Cohen-Or, D.; and Chen, B. 2020. Skeleton-Aware Networks for Deep Motion Retargeting. *ACM Trans. Graph.*, 39(4): 62. 2

Aberman, K.; Shi, M.; Liao, J.; Lischinski, D.; Chen, B.; and Cohen-Or, D. 2019a. Deep Video-Based Performance Cloning. *Comput. Graph. Forum*, 38: 219–233. 2

Aberman, K.; Wu, R.; Lischinski, D.; Chen, B.; and Cohen-Or, D. 2019b. Learning character-agnostic motion for motion retargeting in 2D. *ACM Trans. Graph.*, 38(4): 75:1–75:14. 2, 6

Adobe. 2021. Mixamo. https://www.mixamo.com/. 6

Alldieck, T.; Magnor, M.; Xu, W.; Theobalt, C.; and Pons-Moll, G. 2018. Video Based Reconstruction of 3D People Models. In *CVPR*. 3

Alp Güler, R.; Neverova, N.; and Kokkinos, I. 2018. Densepose: Dense human pose estimation in the wild. In *CVPR*. 3

Artacho, B.; and Savakis, A. E. 2020. UniPose: Unified Human Pose Estimation in Single Images and Videos. In *CVPR*. 3

Bulat, A.; Kossaifi, J.; Tzimiropoulos, G.; and Pantic, M. 2020. Toward fast and accurate human pose estimation via soft-gated skip connections. In *IEEE FG*. 3

Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2018. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*. 3

Chan, C.; Ginosar, S.; Zhou, T.; and Efros, A. A. 2019. Everybody Dance Now. In *ICCV*. 2

Chen, C.-H.; Tyagi, A.; Agrawal, A.; Drover, D.; Stojanov, S.; and Rehg, J. M. 2019a. Unsupervised 3D Pose Estimation with Geometric Self-Supervision. In *CVPR*. 3, 4

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*. 3

Chen, X.; Lin, K.-Y.; Liu, W.; Qian, C.; and Lin, L. 2019b. Weakly-Supervised Discovery of Geometry-Aware Representation for 3D Human Pose Estimation. In *CVPR*. 1, 2

Choi, H.; Moon, G.; and Lee, K. M. 2020. Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose. In *European Conference on Computer Vision (ECCV)*. 1, 6

Ci, H.; Ma, X.; Wang, C.; and Wang, Y. 2020. Locally Connected Network for Monocular 3D Human Pose Estimation. *IEEE TPAMI*, 1–1. 6

Ci, H.; Wang, C.; Ma, X.; and Wang, Y. 2019. Optimizing Network Structure for 3D Human Pose Estimation. In *ICCV*. 1

Deng, B.; Lewis, J. P.; Jeruzalski, T.; Pons-Moll, G.; Hinton, G. E.; Norouzi, M.; and Tagliasacchi, A. 2020. NASA Neural Articulated Shape Approximation. In *ECCV*. 3

Denton, E. L.; and Birodkar, V. 2017. Unsupervised Learning of Disentangled Representations from Video. In *NeurIPS*. 3

Dong, J.; Shuai, Q.; Zhang, Y.; Liu, X.; Zhou, X.; and Bao, H. 2020. Motion Capture from Internet Videos. In *ECCV*. 1, 2

Duan, H.; Zhao, Y.; Chen, K.; Shao, D.; Lin, D.; and Dai, B. 2021. Revisiting Skeleton-based Action Recognition. *CoRR*, abs/2104.13586. 1, 2

Girshick, R.; Radosavovic, I.; Gkioxari, G.; Dollár, P.; and He, K. 2018. Detectron. https://github.com/facebookresearch/detectron. 3

Gleicher, M. 1998. Retargetting Motion to New Characters. In *SIGGRAPH*. 2, 7

Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*. 3

Hodgins, J. K.; and Pollard, N. S. 1997. Adapting simulated behaviors for new characters. In *SIGGRAPH*. 2

Hoshyari, S.; Xu, H.; Knoop, E.; Coros, S.; and Bächer, M. 2019. Vibration-Minimizing Motion Retargeting for Robotic Characters. *ACM Trans. Graph.*, 38. 1

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal Unsupervised Image-to-image Translation. In *ECCV*. 3

Hubert, L.; and Arabie, P. 1985. Comparing partitions. *Journal of Classification*, 2(1): 193–218. 7

Kang, N.; Bai, J.; Pan, J.; and Qin, H. 2019. Real-time Animation and Motion Retargeting of Virtual Characters Based on Single RGB-D Camera. In *IEEE VR*. 1

Kim, T.; and Lee, J. H. 2020. C-3PO: Cyclic-Three-Phase Optimization for Human-Robot Motion Retargeting based on Reinforcement Learning. In *ICRA*. 1

Kim, Y.; Park, H.; Bang, S.; and Lee, S. 2016. Retargeting Human-Object Interaction to Virtual Avatars. *IEEE TVCG*, 22(11): 2405–2412. 1

Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M. 2014. Semi-supervised Learning with Deep Generative Models. In *NeurIPS*. 3

Kocabas, M.; Athanasiou, N.; and Black, M. J. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *CVPR*. 1

Koenemann, J.; Burget, F.; and Bennewitz, M. 2014. Real-time imitation of human whole-body motions by humanoids. In *ICRA*. 1

Lee, H.-Y.; Tseng, H.-Y.; Huang, J.-B.; Singh, M. K.; and Yang, M.-H. 2018. Diverse Image-to-Image Translation via Disentangled Representations. In *ECCV*. 3

Lee, J.; and Shin, S. Y. 1999. A Hierarchical Approach to Interactive Motion Editing for Human-Like Figures. In *SIGGRAPH*. 2

Lian, Z.; Godil, A.; and Xiao, J. 2013. Feature-Preserved 3D Canonical Form. *Int. J. Comput. Vis.*, 102(1-3): 221–238. 3

Lim, J.; Chang, J. H.; and Choi, Y. J. 2019. PMnet: learning of disentangled pose and movement for unsupervised motion retargeting. In *BMVC*. 2

Liu, J.; Shi, M.; Chen, Q.; Fu, H.; and Tai, C.-L. 2021. Normalized Human Pose Features for Human Action Video Alignment. In *CVPR*, 11521–11531. 3

Liu, W.; Piao, Z.; Jie, M.; Luo, W.; Ma, L.; and Gao, S. 2019. Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis. In *ICCV*. 1

Mathieu, M. F.; Zhao, J. J.; Zhao, J.; Ramesh, A.; Sprechmann, P.; and LeCun, Y. 2016. Disentangling factors of variation in deep representation using adversarial training. In *NeurIPS*. 3

Moon, G.; Chang, J.; and Lee, K. M. 2019. Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image. In *ICCV*. 1, 6

Nie, Q.; Liu, Z.; and Liu, Y. 2020. Unsupervised 3d human pose representation with viewpoint and pose disentanglement. In *ECCV*, 102–118. Springer. 3

Novotny, D.; Ravi, N.; Graham, B.; Neverova, N.; and Vedaldi, A. 2019. C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure From Motion. In *ICCV*. 3

Ott, C.; Lee, D.; and Nakamura, Y. 2008. Motion capture based human motion recognition and imitation by direct marker control. In *IEEE-RAS*. 1

Pavllo, D.; Feichtenhofer, C.; Grangier, D.; and Auli, M. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*. 1, 5

Pickup, D.; Liu, J.; Sun, X.; Rosin, P. L.; Martin, R. R.; Cheng, Z.; Lian, Z.; Nie, S.; Jin, L.; Shamai, G.; Sahillioğlu, Y.; and Kavan, L. 2018. An evaluation of canonical forms for non-rigid 3D shape retrieval. *Graphical Models*, 97: 17 – 29. 3

Pickup, D.; Sun, X.; Rosin, P. L.; and Martin, R. R. 2015a. Euclidean-distance-based canonical forms for non-rigid 3D shape retrieval. *Pattern Recognition*, 48(8): 2500 – 2512. 3

Pickup, D.; Sun, X.; Rosin, P. L.; and Martin, R. R. 2016. Skeleton-based canonical forms for non-rigid 3D shape retrieval. *Comput. Vis. Media*, 2(3): 231–243. 3

Pickup, D.; Sun, X.; Rosin, P. L.; Martin, R. R.; Cheng, Z.; Nie, S.; and Jin, L. 2015b. Canonical Forms for Non-Rigid 3D Shape Retrieval. In *Eurographics Workshop*. 3

Rosenberg, A.; and Hirschberg, J. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *EMNLP-CoNLL*. 7

Saito, S.; Simon, T.; Saragih, J.; and Joo, H. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *CVPR*. 3

Shamai, G.; Zibulevsky, M.; and Kimmel, R. 2015. Accelerating the Computation of Canonical Forms for 3D Nonrigid Objects using Multidimensional Scaling. In Pratikakis, I.; Spagnuolo, M.; Theoharis, T.; Gool, L. V.; and Veltkamp, R., eds., *Eurographics Workshop*. 3

Tak, S.; and Ko, H. 2005. A physically-based motion retargeting filter. *ACM Trans. Graph.*, 24: 98–117. 2

Tulyakov, S.; Liu, M.; Yang, X.; and Kautz, J. 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In *CVPR*. 3

Villegas, R.; Yang, J.; Ceylan, D.; and Lee, H. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *CVPR*. 2

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017. Decomposing Motion and Content for Natural Video Sequence Prediction. In *ICLR*. 3

Wandt, B.; Rudolph, M.; Zell, P.; Rhodin, H.; and Rosenhahn, B. 2021. CanonPose: Self-Supervised Monocular 3D Human Pose Estimation in the Wild. In *Computer Vision and Pattern Recognition (CVPR)*. 3

Wang, S.; Geiger, A.; and Tang, S. 2021. Locally Aware Piecewise Transformation Fields for 3D Human Mesh Registration. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 3

Wang, T.-C.; Liu, M.-Y.; Tao, A.; Liu, G.; Kautz, J.; and Catanzaro, B. 2019. Few-shot Video-to-Video Synthesis. In *NeurIPS*. 3

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Liu, G.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. Video-to-Video Synthesis. In *NeurIPS*. 3

Weng, C.-Y.; Curless, B.; and Kemelmacher-Shlizerman, I. 2019. Photo Wake-Up: 3D Character Animation from a Single Photo. In *CVPR*. 3

Yang, W.; Ouyang, W.; Wang, X.; Ren, J.; Li, H.; and Wang, X. 2018. 3D human pose estimation in the wild by adversarial learning. In *CVPR*. 1

Yang, Z.; Zhu, W.; Wu, W.; Qian, C.; Zhou, Q.; Zhou, B.; and Loy, C. C. 2020. TransMoMo: Invariance-Driven Unsupervised Video Motion Retargeting. In *CVPR*. 2, 5, 6, 7

Zhao, L.; Wang, Y.; Zhao, J.; Yuan, L.; Sun, J. J.; Schroff, F.; Adam, H.; Peng, X.; Metaxas, D.; and Liu, T. 2021. Learning View-Disentangled Human Pose Representation by Contrastive Cross-View Mutual Information Maximization. In *CVPR*, 12793–12802. 3

Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; and Li, H. 2019. On the continuity of rotation representations in neural networks. In *CVPR*. 4