Unsupervised 3D Pose Estimation for Hierarchical Dance Video Recognition[*]

Xiaodan Hu, Narendra Ahuja
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
{xiaodan8,n-ahuja}@illinois.edu

Abstract

Dance experts often view dance as a hierarchy of information, spanning low-level (raw images, image sequences), mid-levels (human poses and bodypart movements), and high-level (dance genre). We propose a Hierarchical Dance Video Recognition framework (HDVR). HDVR estimates 2D pose sequences, tracks dancers, and then simultaneously estimates corresponding 3D poses and 3D-to-2D imaging parameters, without requiring ground truth for 3D poses. Unlike most methods that work on a single person, our tracking works on multiple dancers, under occlusions. From the estimated 3D pose sequence, HDVR extracts body part movements, and therefrom dance genre. The resulting hierarchical dance representation is explainable to experts. To overcome noise and interframe correspondence ambiguities, we enforce spatial and temporal motion smoothness and photometric continuity over time. We use an LSTM network to extract 3D movement subsequences from which we recognize dance genre. For experiments, we have identified 154 movement types, of 16 body parts, and assembled a new University of Illinois Dance (UID) Dataset, containing 1143 video clips of 9 genres covering 30 hours, annotated with movement and genre labels. Our experimental results demonstrate that our algorithms outperform the state-of-the-art 3D pose estimation methods, which also enhances our dance recognition performance.

1. Introduction

Dance represents a special genre of human activity. Our goal in this paper is development of algorithms to understand dance videos. We combine estimation of body movements with their feasibility as a part of dance. This enables interpretation of dance videos using not only constraints posed by the data but also those by the domain knowledge.

A variety of proposed methods have also focused on dance videos [1–6]. Most of these rely on kinect sensors to obtain depth information [1, 2]. [3] classifies Indian dances by extracting patches centered at body's joint locations and using an LSTM network for classification. [4] proposes to perform Laban Movement Analysis (in terms of dance domain constructs of Body, Effort, Shape and Space) to then describe human motion from a pose sequence. [5] compares the effects of using three different representations - raw images, optical flow and multi-person pose data - on their proposed dance dataset proving that visual information is not sufficient to classify motion-heavy categories. There are several approaches to action recognition that first estimate poses [7–9]. [7] creates a coaching system for personalized athletic training based on pose correctness. [8] improves action recognition performance by improving pose estimation accuracy using additional spatial and temporal constraints. However, [7, 8] both estimate only the 2D poses, leading to difficulties ambiguity when the movements are along the viewing direction . [9] estimates both 2D and 3D Poses as well as image features to predict actions from all three. [7–9] limit their representation for action recognition to pose sequences without including any higher level semantics that may define action. Moreover, these methods also require pose annotations in training videos. [6] embeds RGB and optical-flow values into a single two-in-one stream network for more efficient dance genre classification. In addition to the features such as pose and optical flow used in these works, in this paper we use dance domain representations to tune feature analysis to dance instead of being generic.

When people dance, they follow a carefully choreographed sequence of 3D *movements*, where each movement is hierarchically composed of simpler movements, ending in *basic movements*. Each basic movement is composed of a sequence of poses representing a specific dance pattern. For brevity, in what follows, we will refer to basic movements by simply movements, We identify movements of 16 main *body parts* $e \in E$ illustrated in Figure 3. following Labanotation [10], a well-known notation system used to record and archive human motion. Then in Table 1 we

[*] The support of the Office of Naval Research under grant N00014-20-1-2444 and USDA National Institute of Food and Agriculture under grant 2020-67021-32799/1024178 are gratefully acknowledged.

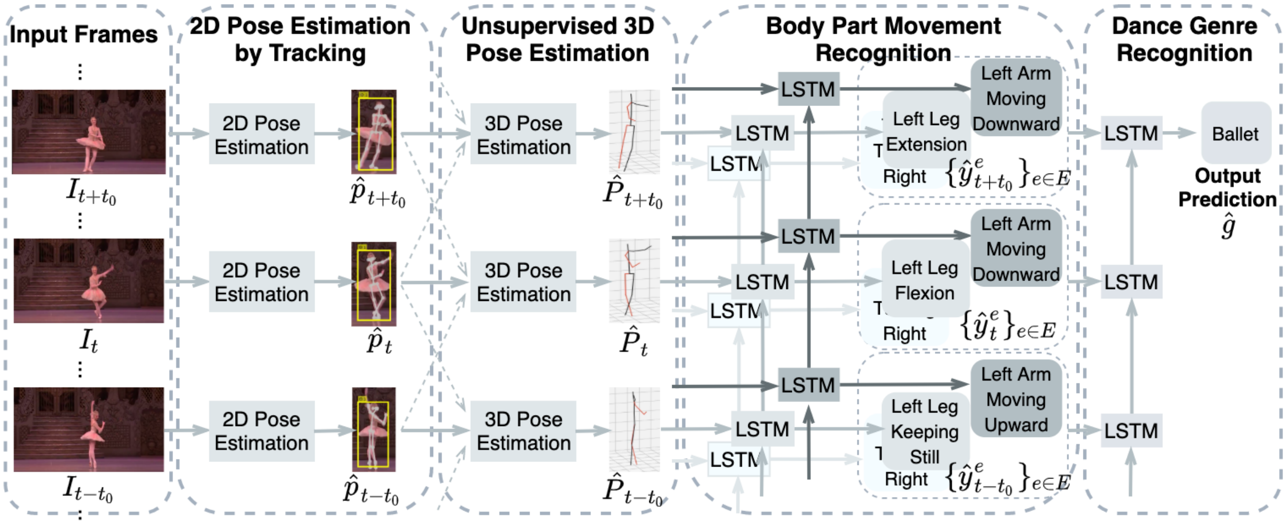


Figure 1. Overview of the model architecture. Given a sequence of video frames $\{I_t\}_{t=0}^{T-1}$, the model analyzes the content in a hierarchical manner, from the low levels (pose estimation & tracking) to the cognitive levels (movement and dance genre recognition). The input sequence $\{I_t\}_{t=0}^{T-1}$ forms the first (bottom) level. At the second level, our algorithm simultaneously estimates the 2D pose $\hat{p}^i_t$ and 3D pose $\hat{P}^i_t$ of each dancer $i(i = 0, ..., N-1)$ at each frame, as well as the camera projection parameters. Our algorithm works under occlusions, e.g., among dancers. At the third level, each dance movement $\hat{y}^e_t$ of each body part $e \in E$ (defined over a sequence of frames) is recognized and its location, given by, e.g., its starting frame t and length are estimated, based on the poses estimated for the previous frames. At the fourth level, the dance genre $\hat{g}$ is recognized based on the movements $\{\hat{y}^e_t\}_{e\in E}$ of all body parts.

list the basic *movements* $y^e \in Y^e$ for each *body part* $e \in E$, again following [10] and defined in terms of homogeneity of motion direction, and level which are frequently used to describe the dance in dance domain. Our dance recognition model adopts this hierarchy used by dance experts, which starts with the 3D pose sequence of the dancer, combines subsequences of joint displacements into dance *movements*, and finally infers dance genre from the sequences of the movements of joints. To help the model segment the pose sequence into the basic movements, we manually annotate the starting and ending positions of such movements for each body part for a subset of videos in the UID dataset. Our framework takes a raw dance video sequence $\{I_t\}_{t=0}^{T-1}$ as input, estimates poses $\hat{p}_t$ for each frame I_t, recognizes the movement $\hat{y}^e_t$ (over multiple frames) of each body part e based on its past pose sequence, and then predicts the dance genre $\hat{g}_t$ from the movement sequence. Experiments show that our hierarchical feature analysis is an effective way to recognize dance and our method outperforms state-of-the-art on F-score.

The main contributions of this paper are as follows:

- We propose the first dance video understanding framework that analyzes the videos hierarchically - from the bottom level of video frames, through the middle level of human poses, to the highest level of movements and associated dance genres.
- Our algorithm tracks and outputs 2D pose of each dancer in each frame in the presence of occlusions among dancers.
- We propose an unsupervised 3D pose estimation algorithm that starts with the estimated 2D pose sequence, and simultaneously and iteratively updates 2D poses, 3D poses and 3D-to-2D projection parameters using a single camera without using ground-truth for these poses or parameters. Our 3D pose network achieves state-of-the-art performance by incorporating kinematic constraints of a 34-DOF human skeletal model and temporal smoothness of motion.
- We have curated a large dance video data set, containing pose in ground truths for each video frame as well as for each movement, which we will share with the community for further exploration.

2. Computational Approach

Figure 1 describes the components of our approach to dance video recognition and the hierarchy they form. Our approach can be summarized in the following steps: Step 1: For each input frame I_t, the model estimates the 2D pose p^i_t for dancer i appearing in I_t. The model tracks approximate locations of the dancers $\{i\}_{i=0}^{N-1}$ throughout the video via their bounding boxes $\{B^i_t\}_{t=0}^{T-1}$. Step 2: At each frame, the model provides an estimate $\hat{p}^i_t$ of the 2D pose p^i_t of the dancer associated with each tracked box B^i_t (Section 2.1). Step 3: The model then estimates 3D poses $\hat{P}^i_t$ from the estimated 2D ones $\hat{p}^i_t$, by using an unsupervised 3D pose

Body Part	Examples of Movement Label	# Labels
Head	Head Turning Up; Head Turning Down; Head Turning Left; Head Turning Right; Head Circling	7
Neck	Neck Moving Left; Neck Moving Right; Neck Circling; Head Keeping Still; Unknown	5
Left Shoulder	Left Shoulder Moving Upward; Left Shoulder Moving Downward; Left Shoulder Circling	5
Left Lower Arm	Left Arm Moving Upward; Left Arm Moving Downward; Left Arm Moving Left	11
Left Upper Arm	Left Arm Moving Upward; Left Arm Moving Downward; Left Arm Moving Left	11
Torso	Torso Bending; Torso Unbending; Torso Turning Left; Torso Turning Right; Torso Swing; Somesault	10
Hips	Hips Waving; Hips Figure 8; Hips Circling; Hip Moving Up; Hip Moving Down; Hips Keeping Still	10
Left Lower Leg	Left Leg Moving Upward; Left Leg Moving Downward; Left Leg Moving Left	15
Left Upper Leg	Left Leg Moving Upward; Left Leg Moving Downward; Left Leg Moving Left	15
Left Foot	Left Foot Extension; Left Foot Flexion; Left Foot Relaxed; Unknown	4

Table 1. Selected examples of movement labels of each body part. To save space, only the movements of the left body parts are shown in the table. The movements of the right body parts are the same as the left ones. There are 16 body parts and 154 movement labels in total.

estimation method (Section 2.2). Step 4: The model uses the LSTM network to recognize the movement $\{\hat{y}_t^e\}_{t=0}^{T-1}$ of each body part $e \in E$ (e.g., head, torso, etc.) from the trajectories $\{\{\hat{P}_t^j\}_{j \in J_e}\}_{t=0}^{T-1}$ of all the joints $j \in J_e$ connected to the body part e, where $J_e \subset E$ (Section 2.3). We represent any given state of a dance as a set of body part configurations and the entire dance as a sequence of such sets. Step 5: For recognition, we first concatenate the movements $\{\{\hat{y}_t^e\}_{e \in E}\}\}_{t=0}^{T-1}$ of all body parts, and input it to an LSTM network to recognize the dance genre $\hat{g}$ (Section 2.4). The rest of this section introduces the components of this hierarchy.

2.1. 2D Pose Estimation by Tracking

```
Algorithm 1: Object Tracking
Input: a sequence of video frames $\{I_t\}_{t=0}^{T-1}$
Output: a sequence of bounding boxes
  $\{(x_t^i, y_t^i, w_t^i, l_t^i)\}_{t=0}^{T-1}$ of the $i^{th}$ dancer
Initialization: select the bounding box
  $(x_0^i, y_0^i, w_0^i, l_0^i)$ of $N$ dancers to track by mouth
while new frame $I_t$ available do
    for $i^{th}$ dancer do
        Obtain $(x_t^i, y_t^i, w_t^i, l_t^i)$ by LDES approach
        if not overlap with others then
            Store histogram and velocity of $i^{th}$
              dancer
        end
        if overlap happens & tracking fails then
            Estimate when overlap ends
        end
        if overlap ends then
            Relocate the bounding box
        end
    end
end
```

To estimate 2D (or 3D) pose, we estimate 2D (or 3D) coordinates of each body joint. Classical pose estimation methods such as pictorial structures framework and deformable part models largely rely on hand-designed features to determine body joint locations. Recently, deep learning-based approaches have achieved a major breakthrough in solving the problems in multi-person pose estimation (e.g., how to group keypoints for different people). They can be divided into top-down [11, 12] and bottom-up [13–15]. The former employ detectors to first locate person instances and then their individual joints; the latter first estimate all joint locations within the image and then assign the joints to the associated person. Although these methods provide superior pose estimates, they have two major shortcomings critical to our task. Firstly, most of the pose estimation methods cannot track a dancer through the video when there are multiple dancers present because they perform pose estimation from individual images, ignoring the temporal information. Besides, the methods perform training mostly on large datasets wherein the dance parts are very small, with a single person, limited pose variety, and clean background. and therefore cannot guarantee accuracy on real world dance videos. The method we propose can track se-

```
Algorithm 2: Tracking Based 2D Pose Estimation
6
Input: a sequence of video frames $\{I_t\}_{t=0}^{T-1}$ and a
  sequence of bounding boxes
  $\{B_t^i\}_{t=0}^{T-1} = \{(x_t^i, y_t^i, w_t^i, l_t^i)\}_{t=0}^{T-1}$ of the $i^{th}$ dancer
Output: a sequence of poses $\{\hat{p}_t^i\}_{t=0}^{T-1}$ of the $i^{th}$
  dancer
while new frame $I_t$ available do
    Estimate poses // Perform OpenPose
    for $i^{th}$ dancer do
        Select pose $\hat{c}$ from $C$ poses overlapped with
          the bounding box $B_t^i$ based on histogram
          match
    end
end
```

lected dancers, detect estimation errors, and correct them automatically.

Object Tracking: As explained in Algorithm 1, our tracking algorithm is built upon the LDES tracker [16]. Since occlusion between dancers is a serious problem, our algorithm centrally addresses it. Following are the three stages of our algorithm: (1) Use the LDES tracker to track each i^{th} dancer when the dancer has no overlap with other dancers, while maintaining a color histogram h_t^i and a bounding box $B_t^i = (x_t^i, y_t^i, w_t^i, l_t^i)$ for the dancer. (2) Detect occurrence of overlap by detecting failure of the tracker as indicated by a significant difference between the directions of motion before and after overlap. (3) Predict the time and the location of the dancer when the overlap may be expected to end, from the location and velocity observed just before the beginning of the overlap. Since multiple dancers may be detected in the vicinity of the predicted location in the predicted frame, select the one that provides the best histogram match, and update h_t^i and B_t^i accordingly.

Tracking Based 2D Pose Estimation: As explained in Algorithm 2, we obtain the initial 2D poses by using the OpenPose method [15]. After we obtain the bounding box B_t^i for each dancer i at the end of the overlap, the box B_t^i may overlap with multiple boxes simultaneously, indicating multiple 2D pose estimation results. We select that pose $\hat{p}_t^i$ whose histogram is most similar to the one $\hat{p}_{t-1}^i$ seen in the previous frame. (Algorithm 2).

2.2. 3D Pose Estimation

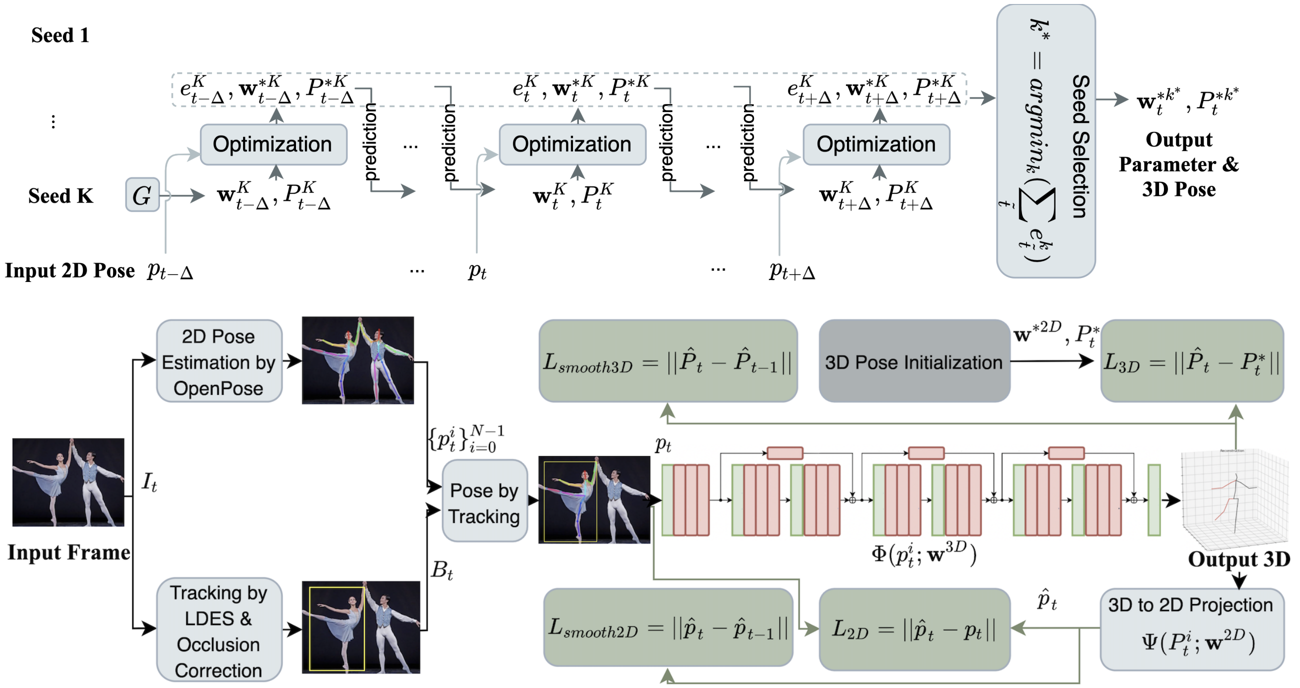
Algorithm 3: 3D Pose Initialization

```
Input: a sequence of 2D poses $\{p_t\}_{t=0}^{N-1}$ of a dancer
Output: a sequence of 3D poses $\{\tilde{P}_t\}_{t=0}^{N-1}$ of the
  dancer
Set the temporal window size to be $2\Delta$
Denote total number of segments as $s = \lfloor \frac{N}{2\Delta} \rfloor$
for $t = \Delta$ to $N - \Delta$ do
    for $k$ = 0 to $K - 1$ do
        Try new seed for DH parameters $\Lambda^k$ and
          perspective projection parameters $\omega^k$
        for $i = t - \Delta$ to $t + \Delta$ do
            Generate 3D pose $\hat{P}_i^k = G(\Lambda^k)$
            Estimate 2D pose $\hat{p}_i^k = \Psi(\hat{P}_i^k; \omega^k)$
            Compute error $e_i^k = ||\hat{p}_i^k - p_i||_2^2$
            Optimize $\Lambda^{*k}, \omega^{*k}$
        end
    end
end
Select the 3D pose corresponding to seed
  $k^* = \underset{\tilde{k}}{\operatorname{argmin}} \sum_{i=t-\Delta}^{t+\Delta} e_i^{\tilde{k}}$ as the initialized pose
```

Algorithm 4: 3D Pose Estimation

```
Input: a sequence of video frames $\{I_t\}_{t=0}^{T-1}$, 2D
  poses $\{p_t\}_{t=0}^{T-1}$ and initial 3D poses $\{\tilde{P}_t\}_{t=0}^{T-1}$ of a
  dancer
Output: a sequence of estimated 3D poses $\{\hat{P}_t\}_{t=0}^{T-1}$
  of the dancer
while new frame $I_t$ available do
    Estimate 3D pose $\hat{P}_t$
    Project to 2D pose $\hat{p}_t$
    Compute loss $L = \alpha(||\hat{p}_t - \hat{p}_{t-1}||_2^2 + \beta||\hat{P}_t -$
      $\hat{P}_{t-1}||_2^2) + ||\hat{p}_t - p_t||_2^2 + ||\hat{P}_t - \tilde{P}_t||_2^2$
    Update $\omega^{2D}$ and $\omega^{3D}$
end
```

Towards our objective of using dance representations close to those used by experts, we need to use 3D, instead of 2D, pose sequences. Similarly for recognition using the language of dance experts, we need to extract descriptors of 3D movements from the 2D pose sequences, which constitute our method's next stage. Computationally too, 3D poses contain more information than 2D poses, and thus lead to more accurate dance recognition. However, predicting 3D poses from 2D poses is an ill-posed problem like other 2D-to-3D problems. The state-of-the-art methods [17–19] use a two-step pipeline for solving it: first detect 2D poses from video frames, and then predict 3D poses by learning the correspondences of 2D and 3D key points. [20] provides a simple yet effective baseline proving that the 2D to 3D task can be solved with a remarkably low error rate. [21] learns a mapping from a distribution of 2D poses to a distribution of 3D poses using an adversarial training approach. However, [20, 21] estimate 3D poses from 2D poses estimated from individual 2D frames, which ignores the temporal continuity information. [22, 23] use temporal correspondences of 2D keypoints to both learn the joint angles as well as predict the joint locations. They compute loss in terms of the distance between these key points and those back-projected using the estimated 3D pose. They enforce such geometric consistency to progressively refine the estimates of 3D poses. However, these methods are based on the assumption that the input 2D poses are accurate. [23] proposes a 2D pose correction module which uses a temporal CNN to refine the 2D initial inputs. However, this assumes that ground-truth 2D poses are available to train the correction module. These assumptions are often restrictive in practice, and do not hold for our dance videos which are collected from the internet. [24] relates detected 2D poses across frames based on tracking-by-detection and then recovers 3D pose in a Bayesian framework. However, their MAP estimation is not robust if the video is long or background changes dramatically. [25] proposes a method to cope with



Seed 1
⋮
Seed K
Input 2D Pose
G
$e^K_{t-\Delta}, \mathbf{w}^{*K}_{t-\Delta}, P^{*K}_{t-\Delta}$
$e^K_t, \mathbf{w}^{*K}_t, P^{*K}_t$
$e^K_{t+\Delta}, \mathbf{w}^{*K}_{t+\Delta}, P^{*K}_{t+\Delta}$
Optimization
prediction
$\mathbf{w}^K_{t-\Delta}, P^K_{t-\Delta}$
$\mathbf{w}^K_t, P^K_t$
$\mathbf{w}^K_{t+\Delta}, P^K_{t+\Delta}$
$p_{t-\Delta}$
p_t
$p_{t+\Delta}$
Seed Selection
$k^* = argmin_k(\sum_t e^k_t)$
$\mathbf{w}^{*k^*}_t, P^{*k^*}_t$
Output Parameter & 3D Pose
2D Pose Estimation by OpenPose
Input Frame
I_t
Tracking by LDES & Occlusion Correction
$\{p^i_t\}^{N-1}_{i=0}$
B_t
Pose by Tracking
p_t
$\Phi(p^i_t; \mathbf{w}^{3D})$
Output 3D
$L_{smooth3D} = ||\hat{P}_t - \hat{P}_{t-1}||$
3D Pose Initialization
$\mathbf{w}^{*2D}, P^*_t$
$L_{3D} = ||\hat{P}_t - P^*_t||$
$L_{smooth2D} = ||\hat{p}_t - \hat{p}_{t-1}||$
$L_{2D} = ||\hat{p}_t - p_t||$
$\hat{p}_t$
3D to 2D Projection
$\Psi(P^i_t; \mathbf{w}^{2D})$

Figure 2. Overview of Proposed 3D pose estimation method. Given a sequence of video frames $\{I_t\}^{T-1}_{t=0}$, the dancers are tracked by our tracking algorithm in Algorithm 1 and each of their 2D poses $\{p^i_t\}^{N-1}_{i=0}$ are estimated by our tracking based 2D pose estimation algorithm in Algorithm 2. Then based on the 2D poses $\{p^i_t\}^{N-1}_{i=0}$, we initialize their 3D poses and camera perspective projection parameters, P^*_t and ω^{*2D}, as shown in Fig. 2 (top) and Algorithm 3. Finally, a neural network is trained to estimate the 3D poses $\{\hat{P}_t\}^{T-1}_{t=0}$, which incorporates kinematic constraints and spatiotemporal smoothness of motion, as described in Algorithm 4.

occlusion. They first infer 3D locations of the visible body joints and then reconstruct the occluded joint locations using learned pose priors and a kinematic skeletal model. [26] fit a parametric human model (SMPL) to observed image key points and segments along with some additional constraints. However, [25, 26] require 3D pose labels and/or shape to supervise the training, which are not available for our "in the wild" video dataset. [27, 28] estimate 3D pose from in-the-wild images without 3D pose annotations, but they require either additional 2D pose datasets or a multi-view setting. To avoid these requirements and the need for groudtruth 2D pose, and to improve computational robustness, we propose an algorithm that integrates 3D pose estimation with 2D pose correction, which can be trained to converge on both estimates simultaneously while also estimating the camera projection parameters consistently.

We use the Denavit-Hartenberg (DH) parameters $\Lambda^k = \{\boldsymbol{\Theta}^k, \boldsymbol{d}^k, \boldsymbol{a}^k, \boldsymbol{\alpha}^k\}$ to represent the 3D pose. A 3D pose $\tilde{P}_t$ is generated by passing Λ^k to the 34-DOF kinematic model G as follows:

$$\hat{P}^k_i = (J_0, J_1, ..., J_{24}) \tag{1}$$

$$J_j = G(\Theta, d, a, \alpha) = \mathcal{T}_\Theta \mathcal{T}_d \mathcal{T}_a \mathcal{T}_\alpha J_{j-1} \tag{2}$$

where

$$\mathcal{T}_\Theta \mathcal{T}_d \mathcal{T}_a \mathcal{T}_\alpha = \begin{bmatrix} \cos\Theta & -\sin\Theta\cos\alpha & \sin\Theta\sin\alpha & r\cos\Theta \\ \sin\Theta & \cos\Theta\cos\alpha & -\cos\Theta\sin\alpha & r\sin\Theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

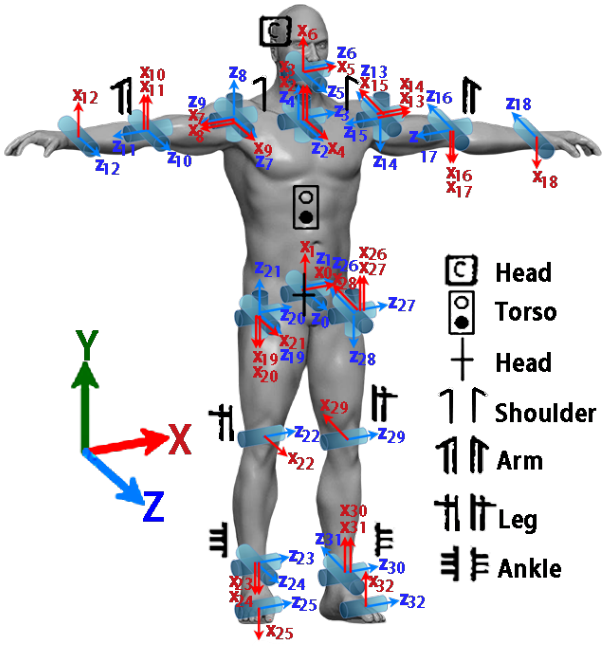


Figure 3. Our 34-DOF digital dancer model. The values of the DH parameters $\Lambda = \{\boldsymbol{\Theta}, \boldsymbol{d}, \boldsymbol{a}, \boldsymbol{\alpha}\}$ of this model are listed in Table 7 in Appendix. The bounds of the joint rotation offset angles $\boldsymbol{\theta}$ and bone length $\boldsymbol{b}$ are defined in Table 8 in Appendix.

where $\mathcal{T}_\Theta, \mathcal{T}_d, \mathcal{T}_a$ and $\mathcal{T}_\alpha J_{j-1}$ are transition matrices, and J_j is the 3D location of the joint j.

We initialize the desired estimates of 3D pose $\tilde{P}_t$ and the 3D-to-2D projection parameter ω_t^{2D} with multiple randomly selected seed pairs $\{\Lambda^{*k}, \omega^k\}$ (to sample the search space), as explained in Figure 2 (top) and Algorithm 3. $\omega^k = \{f^k, c^k\}$ are the perspective projection parameters. At frame t, we sample K seeds of the DH parameters to generate 3D poses $\{\hat{P}_i^k\}_{i=t-\Delta}^{t+\Delta}$ in a sliding window of size 2Δ centered at t and 3D-to-2D projection parameter ω^{2D}. By comparing the reconstructed 2D pose $\hat{p} = \Psi(\hat{P}_t; \omega^{2D})$ projected from the generated 3D pose $\hat{P}_t$ with the input 2D pose p_i estimated in 2.1, we optimize the DH parameters Λ^k generating the 3D pose $\hat{P}_i^k$ while enforcing: (a) constraints that govern the joint rotation offset angles $\boldsymbol{\theta}^k$, (b) consistency with the known bone lengths $\boldsymbol{b}^k$ and (c) temporal smoothness of both the 2D and 3D poses. This is achieved by training with a loss function consisting of two parts: (1) temporal smoothness of both the 2D pose and 3D pose: $\alpha(||\hat{p}_t - \hat{p}_{t-1}||_2^2 + \beta||\hat{P}_t - \hat{P}_{t-1}||_2^2)$. (2) preservation of 3D-to-2D projection (imaging) property: $||\Psi(\hat{P}_t; \omega^{2D}) - p_t||_2^2$. The coefficients α and β are chosen to be inversely proportional to the error: the larger the error, smaller the weight of the window. We also enforce constancy of the 3D to 2D projection parameters by smoothing it over a time window. At each time step t, we update the 3D pose $\hat{P}_t$ and the projection parameter ω^{3D}. From among the solutions obtained using the different seeds, the pair $\{\hat{P}_t^*; \omega^{*2D}\}$ corresponding to the seed offering the least error is selected.

As shown in Figure 2 (bottom), after obtaining the initial 3D pose P_t^* and the 3D-to-2D projection parameters ω_t^{*2D} from the *3D Pose Initialization* block, we train temporal convolutional networks to learn the mapping from the input 2D poses $\{\hat{p}_t\}$ to the 3D ones $\{\hat{P}_t\}$. We use [17] as our baseline networks. During the training, in addition to the consistency between 2D and 3D poses at all times, we again enforce temporal smoothness of motion with the loss function defined as follows:

$$\mathcal{L} = ||\hat{p}_t - p_t||_2^2 + ||\hat{p}_t - p_{t-1}||_2^2 + ||\hat{P}_t - P_t^*||_2^2 + ||\hat{P}_t - \hat{P}_{t-1}||_2^2) \tag{3}$$

where $\hat{p}_t = \Psi(\hat{P}_t; \omega^{*2D})$.See details in Algorithm 4.

To further improve the accuracy when limited labeled 3D ground-truth pose data are available, we introduce a semi-supervised training version of the proposed pose estimation method. A supervised loss is trained by using the available labeled ground truth 3D poses P_t as target, and the loss in Equation (3) is implemented using the remaining unlabeled data. Here, the predicted 3D poses $\hat{P}_t$ are projected back to 2D joint coordinates for consistency with the 2D input p_t. Similar to the training strategy in [17], we jointly optimize the supervised component with our unsupervised component during training, with the labeled data occupying the first half of a batch, and the unlabeled data occupying the second half.

2.3. Body Part Movement Recognition

For each body part e, we train an LSTM-based model to recognize its (basic) movement. During training, the input is a sequence of 3D poses $\{\{\hat{p}_t^j\}_{j \in J_e}\}_{t=0}^{T-1}$ of all the joints $j \in J_e$ connected to the body part e and the output is a sequence of predicted movement labels $\{\hat{y}_t^e\}_{t=0}^{T-1}$ connected to e. Since this is a multi-label classification problem, which means the poses $\{\hat{p}_t^j\}_{j \in J_e}$ connected to the body part e may map to multiple movement labels $\hat{y}_t^e$ of e at the same time, we use the Binary Cross Entropy (BCE) loss between predicted movements $\{\hat{y}_t^e\}_{t=0}^{T-1}$ and the target movement labels $\{y_t^e\}_{t=0}^{T-1}$. This loss is minimized during the training to obtain the optimal model. During testing, the trained model of each $e \in E$ takes a sequence of 3D poses $\{\{\hat{p}_t^j\}_{j \in J_e}\}_{t=0}^{T-1}$ of all the joints connected to e as input, and predicts the movement $\{\hat{y}_t^e\}_{t=0}^{T-1}$ of e.

2.4. Dance Genre Recognition

Analogous to the approach in Section 2.3, we train an LSTM model to take a sequence of movement labels $\{\{\hat{y}_t^e\}_{e \in E}\}_{t=0}^{T-1}$ of all the body parts $e \in E$ as input. We use the output of the last time step from the last layer as the prediction of the dance genre $\hat{g}$. For loss function, we use cross entropy between the predicted dance genre $\hat{g}$ and the target dance genre g. We describe the movement and dance genre recognition in detail in Algorithm 5 and Algorithm 6 in the supplementary document.

3. Experiments

3.1. Data and Experiment Setting

University of Illinois Dance (UID) Dataset. One major challenge for dance recognition lies in the lack of training data. We have curated *UID video dataset* containing 9 types of dances (Ballet, Belly dance, Flamenco, Hip Hop, Rumba, Swing dance, Tango, Tap dance and Waltz) with details listed in Table 2. Figure 4 and 5 show sample frames and information about in our dataset for each dance genre. The videos contain situations of varying difficulty, from simple ones such as tutorial videos with clean background, to hard videos, having interacting dancers, noisy background and varying lights.

Dance Genres	9	**Total Duration**	108,089s
Total # of Clips	1143	**Total # of Frames**	2,788,157
Min clip length	4s	**Min # of clips / class**	30
Max clip length	824s	**Max # of clips / class**	304

Table 2. Summary of the characteristics of the UID dataset.

Evaluation Protocols. we use the widely used mean per-joint position error (MPJPE) in millimeters to calculate the

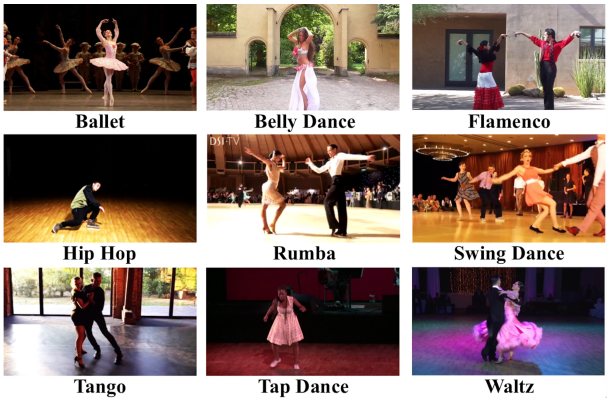


Figure 4. Sample frames for 9 types of dances in the University of Illinois Dance (UID) Dataset.

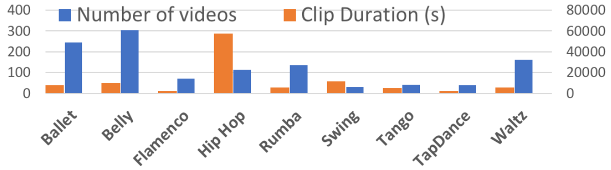


Figure 5. Distribution of the numbers and durations of clips for each genre in the UID Dataset.

Method	Supervision	Extra Data	MPJPE (mm)($\downarrow$)
Martinez [20] ICCV'17	Supervised	-	110.0
Wandt [21] CVPR'19	Supervised	-	323.7
Pavllo [17] CVPR'19	Supervised	-	<u>77.6</u>
Pavllo [17] CVPR'19($\star$)	Semi-Sup.	No	446.1
Ours	Semi-Sup.	No	**73.7**
Zhou [27] ICCV'17	Weakly-Sup.	Yes	93.1
Kocabas [28] CVPR'19	Self-Sup.	Multiview	87.4
Ours	Unsupervised	No	246.4

Table 3. Comparison of 3D pose estimation results using Protocol 1: Mean Per-Joint Position Error (MPJPE) on AIST Dance Video Dataset [29]. ($\star$) uses ground truth 2D poses. Methods using different supervision level are divided by horiontal line. Our proposed method (semi-supervised) achieves the lowest error against the fully supervised methods. Moreover, our unsupervised pose estimation method can achieve the same level of performance as the state-of-the-art supervised/semi-supervised methods.

mean Euclidean distance between the predicted 3D poses $\{\hat{P}_t\}_{t=0}^{T-1}$ and the target 3D poses $\{P_t\}_{t=0}^{T-1}$. We use F-score to measure the accuracy of our movement and dance recognition approaches on our UID dataset.

Experiment Setting. We evaluate our unsupervised 3D pose estimation approach on both the UID video dataset and AIST++ dance dataset [29]. The AIST++ Dataset contains 1,408 multi-view dance sequences from 10 dance genres with hundreds of choreographies, provides 3D human keypoint annotations and camera parameters for 10.1M images, and covers 30 different subjects in 9 views. We did our experiments with a subset of AIST++, containing 200 videos (0.4M frames). 30% of the videos with ground-truth 3D poses are used as labeled data to train the supervised methods [17, 20, 21] and semi-supervised methods ([17] and our method). 10% of the videos are used for testing. The remaining video samples are used as unlabeled data for training the semi-supervised methods.

Method	Supervision	Extra Data	MPJPE (mm)($\downarrow$)
Martinez [20] ICCV'17	Supervised	-	87.3
Zanfir [26] CVPR'18	Supervised	-	69.0
Wandt [21] CVPR'19	Supervised	-	89.9
Pavllo [17] CVPR'19	Supervised	-	**46.8**
Mehta [25] SIGGRAPH'20	Supervised	-	63.6
Pavllo [17] CVPR'19($\star$)	Semi-Sup.	No	51.6
Ours	Semi-Sup.	No	<u>47.3</u>
Zhou [27] ICCV'17	Weakly-Sup.	Yes	64.9
Rhodin [30] ECCV'18	Unsupervised	Multiview	98.2
Kocabas [28] CVPR'19	Self-Sup.	Multiview	60.6
Chen [31] CVPR'19	Unsupervised	Yes	68.0
Kundu [32] ECCV'20	Unsupervised	Yes	67.9
Ours	Unsupervised	No	82.1

Table 4. Comparison of 3D pose estimation results using Protocol 1: Mean Per-Joint Position Error (MPJPE) on Human3.6M Dataset [33] evaluated on S9 and S11. ($\star$) uses ground truth 2D poses. Based on the method's supervision level, five labelled subjects (S1, S5, S6, S7, S8) are used to train the supervised methods, four labelled subjects (S5, S6, S7, S8) and one unlabelled subject (S1) are used to train the semi-supervised methods, and five unlabelled subjects (S1, S5, S6, S7, S8) for the rest methods (e.g., unsupervised). Our proposed method (semi-supervised) achieves the second lowest error against the fully supervised methods. Without the need of additional 2D/3D data, our unsupervised pose estimation method can achieve the same level of performance as the state-of-the-art methods.

For consistency with other work [17, 20, 21], we train and evaluate on $3D$ poses in camera space. In the 3D Pose Initialization component, we use Adam [35] optimizer to optimize the estimated 3D poses in Algorithm 3 for 50 epochs. The temporal window size $\Delta = 3$ and the number of seeds $K = 2$. After obtaining the best initial 3D poses and camera projection parameters (focal lengths and principal points), we use [17] as the baseline to train the 3D pose estimation network for 200 epochs.

3.2. 3D Poses

Figure 6 shows qualitative results of our 3D pose method on both the UID dataset and the AIST++ dataset [29]. The 2D poses (top row) reconstructed from the estimated 3D poses align well with the dancer's movement. The es-

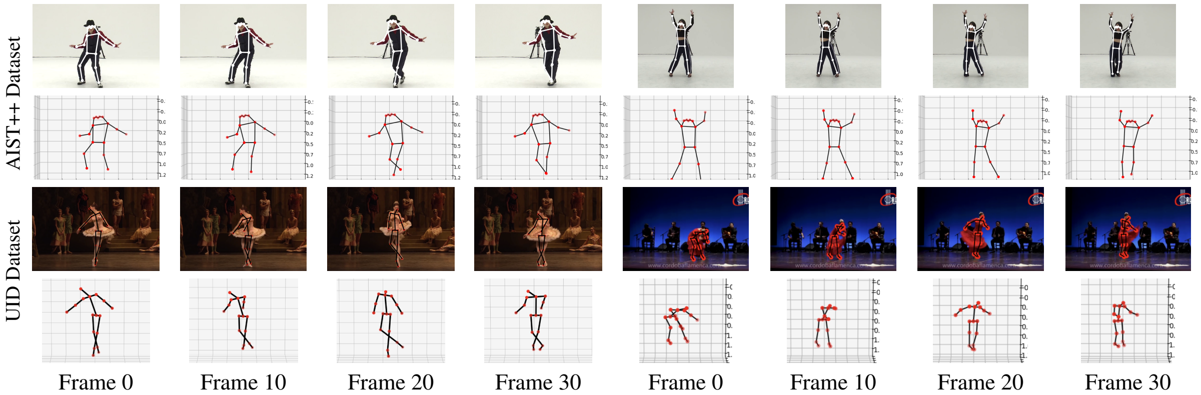


Figure 6. Visualization results on sample videos from the AIST++ dataset [34] and our proposed University of Illinois Dance (UID) dataset. The top row shows the reconstructed 2D poses from the estimated 3D poses and the bottom row shows the estimated 3D poses.

Input to the	F-score											
Movement Recog.	*Averaged*	Head	lshoulder	rshoulder	larm	rarm	Hips	Torso	lleg	rleg	lfoot	rfoot
2D Pose	0.93	**0.95**	**0.96**	**0.96**	0.89	0.91	0.81	0.96	0.94	0.85	**1.00**	**1.00**
3D Pose	**0.97**	0.93	**0.96**	**0.96**	**0.94**	**0.93**	**1.00**	**0.98**	**0.95**	**0.98**	0.99	**1.00**

Table 5. F-scores for body part movements recognition from estimated 2D poses (Sec 2.1) and estimated 3D poses (Sec 2.2) as inputs. Recognition improves as a result of using our estimated 3D poses. Note that the performances for several parts are comparable with existing results. This is because the dancers are at a large distance, diminishing the extra power offered by the 3D information. This situation changes in Table 6.

Input to Dance Genre Recognition	F-score
2D Pose	0.44
3D Pose	0.47
Movements (2D Pose as input)	0.50
Movements (3D Pose as input)	0.55
2D Pose + Movements (2D Pose as input)	**0.73**
3D Pose + Movements (3D Pose as input)	**0.86**

Table 6. Ablation study using different components as inputs. The 3D pose, in general, provides higher accuracy for genre recognition than 2D pose. Combination of the two, 2D and 3D level estimates, achieves better performance than either alone.

timated 3D poses well match the known human skeletal structure and are smooth between frames. To quantitatively evaluate our method, we train our model and three state-of-the-art methods [17, 20, 21] on the AIST++ dataset and calculate the mean per-joint position errors (MPJPE). We also evaluated our model on the Human 3.6M dataset [33]. Table 3 and Table 4 shows that our unsupervised pose estimation method is comparable with the supervised methods. Moreover, our semi-supervised version achieves the best and second best performance on the AIST++ dataset [29] and 3.6M dataset [33], respectively.

3.3. Movement and Dance Genre Recognition

Recognition results for body part movements and dance genre recognition on the UID dataset are given in Tables 5 and 6. We use the 3D poses estimated using our unsupervised method as the input for recognition since our UID collects videos in the wild and hence does not provide ground-truth 3D annotations for training the proposed semi-supervised version. The movements of different body parts can help with dance understanding from the viewpoint of dance experts.

4. Conclusions and Future Work

In conclusion,we have presented an approach to dance videos understanding that follows a hierarchical representation used by experts to describe dances. We have presented an approach to extract the primitives occurring at each level of the representation, from raw videos, to 3D pose, to movements, to dance genre. We have presented the challenges we have encountered and how we have addressed them using new constraints and algorithms. Note that the training in our current dance video recognition framework is not fully unsupervised. We plan to develop a fully unsupervised pipeline that could be jointly trained for pose estimation and genre recognition. In addition, we plan to synthesize dances using the representations we have extracted. We also plan to use the judgments of expert viewers on the quality of the synthesized dance videos as qualitative metrics of the representations extracted by our algorithms.

References

[1] Eftychios Protopapadakis, A. Grammatikopoulou, Anastasios Doulamis, and Grammalidis Nikos. Folk dance pattern recognition over depth images acquired via kinect sensor. *ISPRS*, XLII-2/W3:587–593, 02 2017. 1

[2] H. Matsuyama, K. Hiroi, K. Kaji, T. Yonezawa, and N. Kawaguchi. Hybrid activity recognition for ballroom dance exercise using video and wearable sensor. In *ICIEV and icIVPR*, 2019. 1

[3] Swati Dewan, Shubham Agarwal, and Navjyoti Singh. A deep learning pipeline for Indian dance style classification. In *ICMV*, volume 10696, 2018. 1

[4] S. Dewan, S. Agarwal, and N. Singh. Spatio-temporal laban features for dance style recognition. In *ICPR*, 2018. 1

[5] Daniel Castro, Steven Hickson, Patsorn Sangkloy, Bhavishya Mittal, Sean Dai, James Hays, and Irfan A. Essa. Let's dance: Learning from online dance videos. *CoRR*, 2018. 1

[6] Jiaojiao Zhao and Cees G. M. Snoek. Dance with flow: Two-in-one stream action detection. In *CVPR*, June 2019. 1

[7] Chunyu Wang, Yizhou Wang, and Alan L. Yuille. An approach to pose-based action recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2013. 1

[8] Jianbo Wang, Kai Qiu, Houwen Peng, Jianlong Fu, and Jianke Zhu. Ai coach: Deep human pose estimation and analysis for personalized athletic training assistance. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019. 1

[9] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[10] Ann Hutchinson Guest. *Labanotation: The System of Analyzing and Recording Movement*. Routledge, 4th Edition (February 15, 2005). 1, 2

[11] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *PAMI*, pages 1–1, 2020. 3

[12] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018. 3

[13] S. Kreiss, L. Bertoni, and A. Alahi. Pifpaf: Composite fields for human pose estimation. In *CVPR*, 2019. 3

[14] Bowen Cheng, Bin Xiao, Jingdong Wang, Humphrey Shi, Thomas Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. 2020.

[15] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *PAMI*, 2019. 3, 4

[16] Yang Li, Jianke Zhu, Steven C.H. Hoi, Wenjie Song, Zhefeng Wang, and Hantang Liu. Robust estimation of similarity transformation for visual object tracking. In *AAAI*, January 2019. 4

[17] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 6, 7, 8

[18] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei. 3d human pose machines with self-supervised learning. *PAMI*, 42(5):1069–1082, 2020.

[19] Muhammed Kocabas, Nikos Athanasiou, and Michael Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 4

[20] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 4, 7, 8

[21] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4, 7, 8

[22] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation in videos, 2020. 4

[23] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *CVPR*, pages 896–905, 2020. 4

[24] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Monocular 3d pose estimation and tracking by detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 4

[25] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D motion capture with a single RGB camera. volume 39, 2020. 4, 5, 7

[26] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 7

[27] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5, 7

[28] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5, 7

[29] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, pages 501–510, Delft, Netherlands, November 2019. 7, 8

[30] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation learning for 3d human

pose estimation. In *ECCV*, 2018. 7

[31] C. Chen, A. Tyagi, A. Agrawal, D. Drover, R. MV, S. Stojanov, and J. M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[32] Jogendra Nath Kundu, Ambareesh Revanur, Govind V Waghmare, Rahul M Venkatesh, and R Venkatesh Babu. Unsupervised cross-modal alignment for multi-person 3d pose estimation. 2020. 7

[33] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014. 7, 8

[34] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021. 8

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 7