A Parametric Approach for Generating and Estimating 3D Quadruped Poses

DU Yinwei, ZHANG Ziyan
The Hong Kong University of Science and Technology
{ydual,zzhangcm}@connect.ust.hk
Chi-Keung Tang
The Hong Kong University of Science and Technology
cktang@cs.ust.hk
Yu-Wing Tai
Tencent
yuwingtai@tencent.com

Abstract

Videography provides valuable data for studying animal behaviors where noninvasive tracking of animals is the essential first step. This paper introduces a parametric approach for generating a massive synthetic dataset for quadrupeds containing more than 10 million ground-truth 3D poses with corresponding 2D landmarks. The generated poses are valid since they are constrained by a hierarchical skeletal model. Our dataset for single-image 3D pose estimation can be readily extended to videos. Two novel network architectures are proposed for 3D pose estimation, one for single images and the other for videos. We demonstrate previously unseen and strong results on tracking quadrupedal animals movements in video sequences while estimating the corresponding 3D pose sequences. We present quantitative evaluation and ablation studies, and further demonstrate our approach's generalizability by estimating 3D poses in significantly different body configurations, and vastly different postures where not all four legs are on the ground.

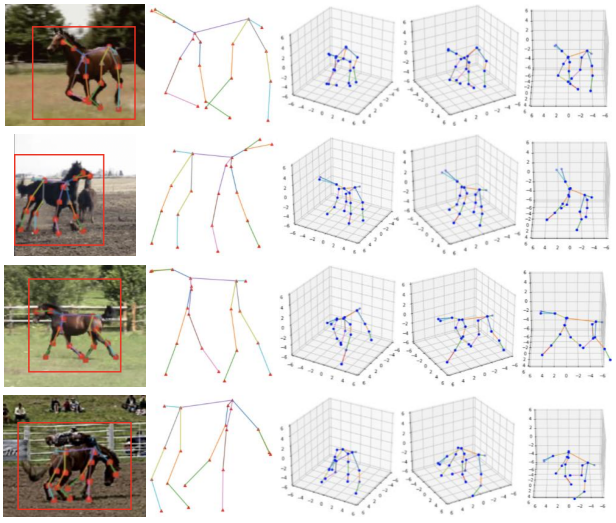

Figure 1. 3D quadruped pose estimation from 2D landmarks of a running/walking horse. The estimated skeletons are overlaid on the input frames. For each example, we also show the estimated 3D skeleton at the same and three different camera viewpoints. See supplemental videos.

1. Introduction

Pose estimation in mainstream computer vision research has almost exclusively focused on humans with no significant or representative work on animal pose estimation from images/videos. In neuroscience and beyond, understanding animal behaviors is an important area where analyzing and hence tracking animal movements is the first necessary step. This step is only possible through noninvasive behavioral tracking and poses extraction [30]. Historically, manual data collection and preparation for analyzing animal movements was prohibitively inefficient, requiring time-consuming and labor-intensive manual procedures. All the known datasets relevant to animal pose estimation are small [8].

This paper focuses on quadrupeds (four-legged animals) and contributes the first principled approach for animal pose estimation from single images and videos, Figure 1. Unlike previous deep learning methods which return 3D coordinates (heat maps) of each joint (e.g. [31, 36]), our parametric approach estimates joint angle and length which are constrained by our hierarchical model to guarantee their validity in the animal pose space. Two novel network archi-

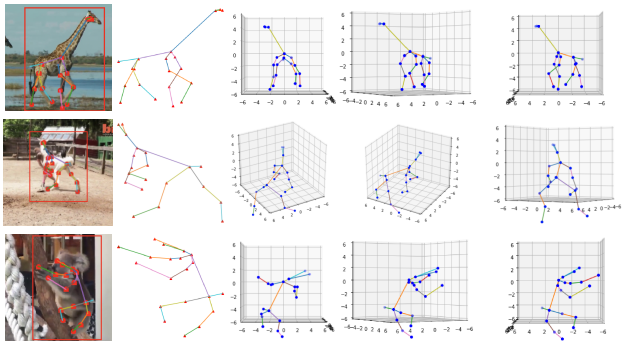

Figure 2. Our approach is generalizable to animals of significantly different dimensions (giraffe with long neck), postures and appearance (alpaca and koala). Same figure arrangement as Figure 1. Testing videos above are from YouTube. See supplemental videos.

tectures are proposed for 3D pose estimation, one for single images and the other for videos. This parametric approach also enables the easy generation of our massive data set which consists of more than 10 millions ground-truth 3D poses with corresponding 2D landmarks. Thus, sufficient training data is available for a deep network to learn how to resolve complex pose ambiguities and self-occlusions due to 2D projections, which is further aided by exploiting temporal consistency inherent video data. Note further that unlike human pose data capture where markers are typically used on real humans, our generative approach is entirely markerless, an essential feature for capturing poses of deadliest or tiny animals. In summary, our contributions are

- a new solution for 3D animal pose estimation from images/videos using a 3D parametric representation for animals, a simple algorithm for synthetic data generation, and new deep network architectures with temporal coherence consideration;
- the first 3D pose estimation algorithm without human-captured 3D landmark annotation; our results on public annotated and our self-labeled videos show that our method is successful in predicting animal poses in presence of complex pose ambiguities and self-occlusion.

With our parametric approach and large-scale dataset, we can readily transfer tracking pose estimation to animals with significantly different body configurations and postures. See Figure 2.

2. Related Work

In the following we review related work in 2D and 3D human pose estimation and pertinent datasets, followed by reviewing the sparse set of papers and datasets in biological and anatomical science on animal pose estimation.

2.1. Human Pose Estimation

2D Pose Estimation. With the goal of predicting a set of 2D joint coordinates from given images, deep learning on pose estimation was first proposed in [39] where convolutional neural network was used as a building block for regressing different body joints. An efficient position refinement model was proposed in [38] where heat-maps rather than specific joint positions are predicted. Convolutional pose machine [43] was proposed which consists of an image feature computational module along with a prediction module. A self-correcting model was used in [5] that progressively refines the initial prediction by feeding back error results. The well-known landmark detector [31] performs repeated bottom-up, top-down processing with intermediate supervision. To simplify existing human pose estimation models, in [44] the authors presented an approach based on the Res-Net appended with a number of deconvolutional layers. In a more recent work [32], a high-resolution representation was maintained, achieving the state-of-the-art in key-point detection and single/multi-person pose estimation task on the COCO dataset [25].

3D Pose Estimation. 3D pose estimation is more challenging than the 2D counterpart. Multi-task learning framework was used in [24] for joint point regression and joint point detection which disentangles the dependencies among different body parts and learns their correlations without explicit constraints. In [7], numerous 2D projections were generated based on a collection of 3D poses with the depths of an input image estimated by traversing a large 3D pose library. Weakly-supervised transfer learning [46] uses 2D and 3D labels as input to a two-stage cascaded structure for estimating human poses. A simple integral operation was used in [36] to relate and unify heat map representation and joint regression to avoid non-differentiable post processing and quantization error. Latest research has focused on multi-person pose estimation from different viewpoints [10], or exploiting weakly-supervised training to analyze images in the wild [42].

Human Pose Datasets. Common pose datasets such as MPII [4], LSP [21], PoseTrack [3] and FLIC [33] are 2D human pose datasets with annotated body joints. Each of them contains around only 10K images. Human3.6M [20] contains 3.6 millions 3D human poses and their corresponding images captured from 17 professional actors. Other notable 3D human datasets include HumanEva [35], Monocular 3D [28], Unite the People [23], DensePose [16], SURREAL [41] and VGG [6]. None of these datasets however is on par with our 10-million animal pose dataset in terms of the number of available poses and body configurations. It is challenging to create a large 3D dataset because of large amount of human annotation, special capture hardware and environment. Furthermore, techniques applied to human cannot be easily transferred to animals, and a large 3D *real*

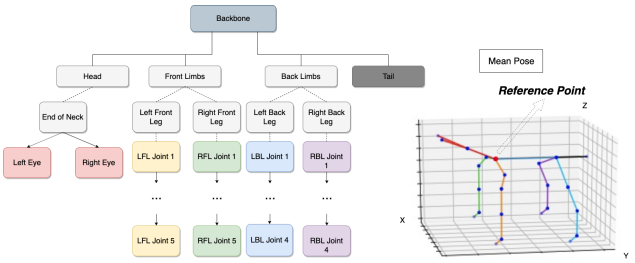

Figure 3. The hierarchical model contains 24 joints roughly in 5 parts: head, front left limb, front right limb, back left limb and back right limb. The location of each joint is retrieved from the relative position with its parent node according to the offset.

animal dataset is even harder to capture because these animals can be tiny and even deadly.

2.2. Animal Pose Estimation

The DeepLabCut toolbox [27] was used by biologists for automatically marking feature points on animals (e.g., mouse head) in a video clip in order to learn their behavioral patterns. They used transfer learning between human pose estimation and animal pose estimation with an appropriate initial labeled training set. Compared to our work, their toolbox still requires a large amount of human annotation, and most of their videos were taken under special laboratory setting and environment.

A number of applications was proposed in their subsequent work [30] including analyzing 3D animal kinetics for horses and cheetahs. Although they also used a rough animal skeleton for analysis, such prediction was based on pictures taken from six different cameras and had very limited set of joints on the graph. Compared to their results, we use a homogeneous skeleton for quadrupedal mammals and produce better results.

Animal Pose Dataset. Most common datasets such as ImageNet [9] and YouTube8M [1] contain labeled animals images/videos but they lack the necessary joint location information. Tigdog [8] is the only dataset to our knowledge with annotated landmarks but its size is not comparable to the above-mentioned human pose datasets. To generate an animal pose dataset that is sufficiently complete (sufficient coverage in the articulated motion space) and compact (no invalid poses), applicable range of motion for each joint should be applied. Different animals have different valid ranges, and we refer readers to pertinent anatomical works on quadrupedal mammals, such as the range of head-neck movement [14], arm glenoid line [34] and knee joint [22], which guide our design setting on the parameter range when we construct our synthetic animal pose dataset.

3. Animal Pose Estimation from Single Images

In this section, we will first describe our parametric skeletal model for single pose generation (section 3.1), which not only is useful for creating data but also speeds up the training process, since we can directly regress on the skeletal parameters and avoid the time-consuming back-projection to the 2D space. Then, we will describe our synthetic dataset for single images (section 3.2) and finally 3D pose estimation network (section 3.3). This single-image framework will be extended to process videos (section 4).

3.1. Parametric Skeletal Model

Following conventional 3D human pose estimation, we adopt a similar hierarchical skeletal representation used in Human 3.6M [20] where relative 3D joint positions are used. Thus, given one pose, a set of offsets is used to define their respective 3D locations.

Single Pose Generation. Our animal skeletal model is a tree structure encompassing 24 joints. Figure 3 shows the skeletal model, which is rooted at the front of backbone, with one end as neck (the reference point shown) and the other end as hip. The final skeleton $\mathbf{P}_{3D}$ can be expressed as:

$$\mathbf{P}_{3D} = f(\mathbf{O}) \tag{1}$$

where f denotes the hierarchical function, and $\mathbf{O}$ is the joint-wise offsets consisting of 23 pairs of spherical parameters:

$$\begin{aligned} \mathbf{O} &= (O_0, O_1, ..., O_{22}) \\ O_i &= (r_i, \theta_i, \phi_i) \end{aligned} \tag{2}$$

where r, θ, ϕ are the (spherical) coordinates with respect to its parent node. With the hierarchical model, given the 3D coordinates (x_p, y_p, z_p) of the parent joint, the 3D coordinates of its child joint can be determined as:

$$\begin{aligned} x_i &= x_p + r_i \sin\theta_i \cos\phi_i \\ y_i &= y_p + r_i \sin\theta_i \sin\phi_i \\ z_i &= z_p + r_i \cos\theta_i \end{aligned} \tag{3}$$

which can be applied recursively in the tree structure. The 2D position $\mathbf{P}_{2D}$ for each joint can be obtained by an orthographic camera projection p and 3D rotation matrix A:

$$\mathbf{P}_{2D} = p(\mathbf{P}_{3D}, A) \tag{4}$$

We take occlusion into consideration by adding a confidence indicator $c_i = \pm 1$ for each pair of 2D coordinates (or landmarks). For a given joint i, $\mathbf{P}_{2D}[i]$ is arbitrarily set if $c_i = -1$ (occlusion).

To define A, in our implementation, we fix the camera and rotate the skeleton (the other equal alternative is fixing the skeleton and rotating the camera). Refer to Figure 3; we

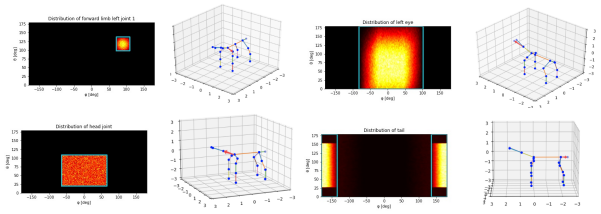

Figure 4. Range and distribution of our synthetic animal data. Red arrow in the 3D model indicates the corresponding joint.

define the local coordinate system centered at the reference point of the skeleton where the x-axis is along the backbone in the mean pose and the z-axis is along one of the vertical legs. Let α, β, γ be the respective rotation angles about the x, y, z-axis. We denote $A = R_z(\gamma)R_y(\beta)R_x(\alpha)$ as the rotation matrix of the 3D skeleton.

During our experiments we found that providing a rough initial rotation estimation $\hat{A}$ can greatly improve prediction accuracy. We also note that the edges of backbone-tail (denoted as bt), backbone-left_forward_joint_0 (denoted as $blf0$) and backbone-right_forward_joint_0 (denoted as $brf0$) are good references to estimate the rotation. Using $\mathbf{v}$ as the vector representation of these edges, we have: $\alpha = \arccos\left((\mathbf{v}_{blf0} + \mathbf{v}_{brf0}) \cdot \mathbf{y}/||\mathbf{v}_{blf0} + \mathbf{v}_{brf0}||\right)$, $\beta = \arccos\left(\mathbf{v}_{bt} \cdot \mathbf{x}/||\mathbf{v}_{bt}||\right)$ and $\gamma = \arccos\left(r_{bt}/r_{ref}\right)$, where $\mathbf{x}$ and $\mathbf{y}$ are respectively the x, y axes of the local coordinate system, and r_{ref} is a normalized reference length that depends on the animal species.

If some of these reference edges are occluded or their reprojection errors are too large, we assign $\hat{A}$ a set of fixed and discrete values as the initial rotation and choose the best results output by the network.

3.2. Synthetic Pose Dataset

To make our synthetic pose distribution as close as possible to the real-world distribution, we empirically define common poses such as standing, walking, running, jumping and lying. Each of them has a distinct range of motion, and poses are generated randomly within the allowable range at each tree node. However, as our articulated poses may not be exhaustive to cover all possible poses, we additionally define a much larger range of motion and then randomly generate poses within to enrich our dataset. The size of random poses is kept relatively small as they may introduce unwanted noise to the dataset.

Completeness and Compactness. Figure 4 shows the valid range and distribution (with respect to θ, ϕ) for selected joint connections. An ideal synthetic dataset should be both complete (include all possible valid poses) and compact (exclude all invalid poses). Indeed, we cannot build a dataset with absolute completeness and compactness (also true for human pose estimation which suffers from extreme poses [2]). In this paper, we argue that we achieve both

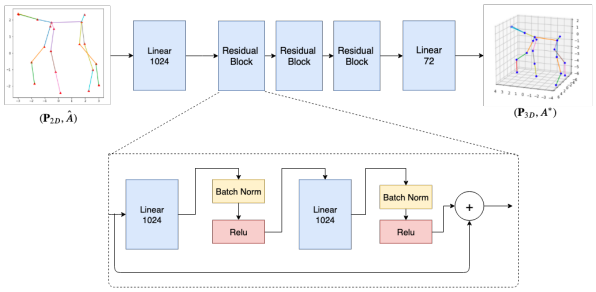

Figure 5. Our 2D-to-3D network. $(\mathbf{P}_{2D}, \hat{A})$ and $(\mathbf{P}_{3D}, A^*)$ are respectively the concatenated input and output of the network as described in the text.

completeness and compactness to a high degree by: 1) categorizing animal movements and adopting parameters from real anatomy data; and 2) allowing a small portion of data to be generated with pure randomization.

3.3. 2D Landmarks to 3D Skeleton

3.3.1 Pose and Rotation Estimation

Our network predicts both 3D pose $\mathbf{P}_{3D}$ and rotation A at the same time. Figure 5 shows the main component of the estimation network which consists of three cascaded residual blocks, where each block has two fully connected layers each of size 1024 followed by batch normalization [19] and RELU activation [29]. In addition, one fully connected layer is applied before the residual blocks to increase the input dimensionality to 1024, and the other one is applied before the final prediction to reduce the size to $72 = 24$ joints $\times$ 3. The input to the network is a 75 dimensional vector consisting of flattened 2D pose $\mathbf{P}_{2D}$ (of size 72) and a rough rotation $\hat{A}$ (3 rotation angles); the network then outputs a 72 dimensional vector containing the 3D offset that can be reshaped to $\mathbf{O}$ (of size 69) and a 3D rotation matrix A^* (of size 3). Ablation studies and comparisons with other block architectures can be found in section 5.

3.3.2 Losses

With our parametric representation, we can easily define the following losses to guarantee only valid poses are generated and ambiguities are resolved:

Symmetry of limbs. Due to the symmetrical structure of animals, the left and right limbs should have the same length, which can be translated into:

$$\mathcal{L}_\ell = \sum_{(i,i')\in\ell} (r_i - r_{i'})^2 \tag{5}$$

where ℓ is the subset of symmetric joint pairs (e.g., left and right eyes, left and right limbs) and (i, i') are the corresponding left and right body parts.

Ambiguity of rotation angles. The rotation matrix A is composed of three angles α, β, γ. However, given any α, $\alpha + 2k\pi$ produces the same effect as α, so we cannot directly compare the angle value. Instead, we use trigonometric functions to preserve continuity:

$$\mathcal{L}_c = \sum_{\delta \in \alpha,\beta,\gamma} (\sin\delta - \sin\delta')^2 + (\cos\delta - \cos\delta')^2 \quad (6)$$

Final loss function. Combining the above and the joint-wise offset difference, the final loss function is:

$$\mathcal{L} = (\mathbf{O} - \mathbf{O}')^2 + w_\ell \mathcal{L}_\ell + w_c \mathcal{L}_c \quad (7)$$

where $\mathbf{O}$ are the ground-truth offsets and $\mathbf{O}'$ are the predicted offsets, and w is the hyper parameter.

4. Animal Pose Estimation from Videos

In this section, we will first present the overall video framework (section 4.1) which consists of our temporal module implemented using temporal convolutional network (section 4.2) on top of single image predictions. The temporal module can effectively deal with occlusions in real animal videos, such as [8], a labeled dataset that will be used in our evaluation. Finally, we will describe our synthetic video dataset generation process (Section 4.3).

Figure 6 shows that in our dataset a given 2D projection can correspond to multiple valid 3D skeletons. This ambiguity can be solved by taking into account neighboring frames, that is, estimating 3D pose from videos.

4.1. Overall Video Framework

Our overall video framework of 3D animal pose estimation makes uses of the single frame module (described in section 3.3) and video module (section 4.2 to be described). Specifically, given a N-frame video sequence with per-frame 2D landmarks $\mathbf{P}_{2D}$, we first process each frame separately as described section 3.3. Then we choose the best

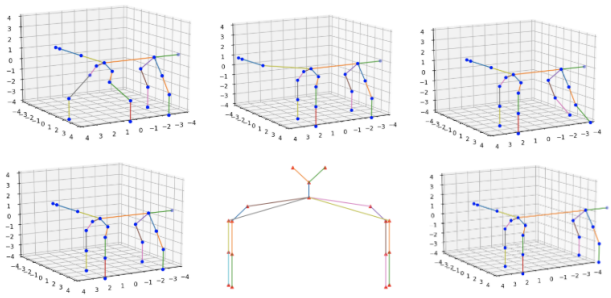

Figure 6. Unlike other animal datasets which contain only a subset of all the possible poses, ours are more complete: we discover multiple valid 3D skeletons (above) can project to the identical 2D landmarks.

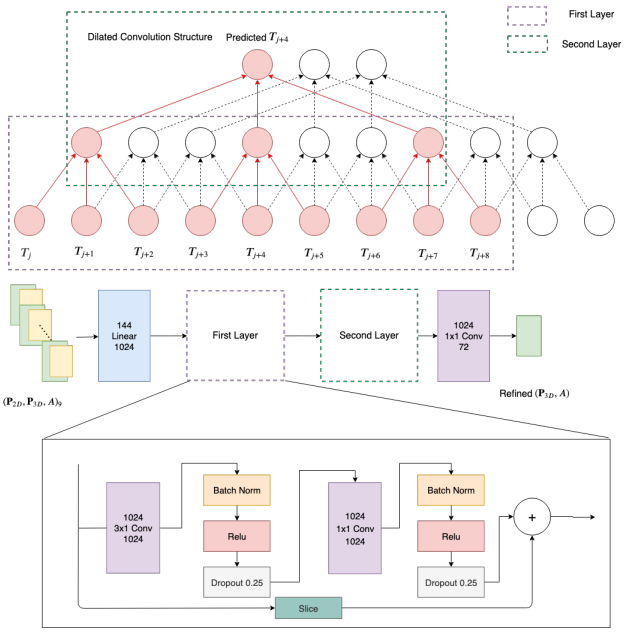

Figure 7. Our temporal network. Each output refined skeleton $(\mathbf{P}_{3D}, A)$ is predicted from 9 sets of input 2D+3D data denoted as $(\mathbf{P}_{2D}, \mathbf{P}_{3D}, A)_9$. Red lines illustrate how prediction for one frame is achieved with its detailed network architecture shown in the bottom.

rotation A and apply it to all N 3D skeletons. This step is useful because within the short period concerned ($N = 30$ in our videos), the relative camera-skeleton position can be assumed to stay stable and thus rotation A can be used; otherwise the result will be jittery which will adversely affect the next module. Finally, we input the N individual skeletons with their corresponding 2D landmarks and rotation A into our temporal network to generate the 2D and 3D skeleton sequence with temporal coherence consideration where, as will be demonstrated in the experimental section, occlusions and pose ambiguities are largely resolved. The above can readily be scaled to longer videos with $N > 30$.

4.2. Temporal Network Module

In real video sequences, some joints may disappear due to self-occlusion or occlusion by other objects and then reappear afterward. Thus taking consideration of multiple frames in proximity can help determine the 3D location of the occluded points.

Although no annotation is needed and recent results have greatly improved, optical flows [26, 12] still may not work well in real videos. While recurrent neural network [37] and long short-term memory [13] have demonstrated success in processing sequential multimedia data and has been widely used in speech recognition [15], video processing [40] and text analysis [45], they suffer from problems such as vanishing gradients. More recently, temporal convolution networks [32] have achieved excellent accuracy in 3D hu-

man pose estimation with relatively fewer parameters, faster training and inference. In our framework, we adopt a similar network architecture to exploit temporal coherence inherent in animal videos.

Refer to Figure 7: given a N-frame video, the input (N sets of 2D landmarks $\mathbf{P}_{2D}$ and predicted 3D skeleton $\mathbf{P}_{3D}$ with rotation A) will be first rearranged into N blocks, where each block contains the data of 9 frames around a given frame and is denoted as $(\mathbf{P}_{2D}, \mathbf{P}_{3D}, A)_9$. For example, in the upper half of the figure, if we want to get the refined 3D skeleton of the frame T_{j+4}, the corresponding input block is a concatenation of $(\mathbf{P}_{2D}, \mathbf{P}_{3D}, A)$ from T_j to T_{j+8}.

The bottom half of Figure 7 shows the network architecture which mainly contains two ResNet-style layers. In each layer, a 3×1 convolution with proper dilation (1 for the first layer and 3 for the second layer) is first performed and then followed by another 1×1 convolution. Each convolution block is followed by batch normalization [19], RELU activation [29] and a dropout [17] of 0.25. An additional convolution block at the end reduces the dimensionality from 1024 to 72 to match $(\mathbf{P}_{3D}, A)$. We do not use more hidden layers because typical video clips in existing animal video data sets such as [8] are of very limited length. Padding provides little information and may eventually harm the prediction performance.

4.3. Video Dataset

We generate video training data in order to exploit temporal coherence. One plausible approach to synthesize video training data is first defining motion sequences for each joint and then calculating the corresponding positions in adjacent frames. Yet, this method is time-consuming and needs prior knowledge about animal movements. Besides, as joints are interdependent and children nodes will inherit the motion of parent nodes, the angular speed will be a complex function with respect to different joints.

Instead, we adopt a trajectory-based approach to synthesize our video training data. Suppose $\mathbf{P}^1_{3D}, \mathbf{P}^2_{3D}$ and $\mathbf{P}^3_{3D}$ are three animal skeletons. The final time-dependent pose $\mathbf{P}_{3D}(n)$ is a linear interpolation among $\mathbf{P}^{1,2,3}_{3D}$, which can

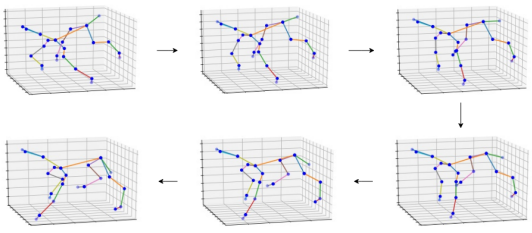

Figure 8. Example of our animal video sequence (jumping). For ease of visualization we show the result of every three frames.

be written as following (n is the current frame index and N is the total number of frames we want to generate):

$$\begin{aligned}\mathbf{P}_{3D}(n) =&(1 - n\alpha/N)\mathbf{P}^3_{3D} + \alpha(t_{1_{n-1}} + \Delta_{t_{1_n}})\mathbf{P}^1_{3D} + \\ &\alpha(t_{2_{n-1}} + \Delta_{t_{2_n}})\mathbf{P}^2_{3D} \\ t_{(1,2)_n} =&\sum_i^n (\Delta_{t_{(1,2)_i}}), \quad \Delta_{t_{(1,2)_i}} \sim \mathcal{N}(\frac{1}{2N}, \sigma^2)\end{aligned} \tag{8}$$

Let A_1 and A_2 be two rotations with small variation, and the time-dependent rotational parameter $A(n)$ is a linear interpolation between A_1 and A_2:

$$A(n) = (1 - n/N)A_1 + nA_2/N \tag{9}$$

The corresponding 2D coordinates $\mathbf{P}_{2D}(n)$ is a function of $\mathbf{P}_{3D}(n)$ and $A(n)$:

$$\mathbf{P}_{2D}(n) = p(f(\mathbf{P}_{3D}(n), A(n))) \tag{10}$$

Figure 8 shows sample videos generated using the above method.

In practice, we adopt several poses (5 to 9) to synthesize a relatively coherent video sequence. The reasons for using multiple poses interpolation and random variables are to imitate real animal movements at non-constant speed and to provide a larger diversity and more degrees of freedom.

5. Experimental Analysis and Results

In this section, we will first present quantitative and qualitative evaluation on animal pose estimation from single images (section 5.1) and then videos (section 5.2), followed by a systematic ablation study on losses and network architecture (section 5.3). Refer to our supplemental videos.

We test our framework on three types of data: our synthetic dataset, existing animal datasets such as [8] and our manually labeled videos which contain a variety of animals such us cat, koala, and giraffe (Figure 2). For existing animal datasets, they already include annotations so we just remap their landmarks to match our defined model. As for manually labeled data, we first convert the video into a sequence of frames and then use photo-editing tool to label each frame individually.

We use the mean per joint offset error as the evaluation metric for synthetic data, and the 2D mean per joint position error (or simply reprojection error) for real data with annotations, which are consistent with the evaluation on human pose estimation. Our experiments were conducted on Intel Xeon E5-2650 v4 and GeForce GTX 1080Ti.

5.1. Images

This section evaluates the results of our proposed single image module, 2D-to-3D network. Given an animal image with annotated landmark positions, we first preprocess the

input data by normalizing all inputs to lie within a working range. The output from the network will be scaled and translated back.

In our experiments, if rotation $\hat{A}$ cannot be determined due to occlusion, we set $\alpha \in [-0.1\pi, 0.1\pi]$, $\beta \in [-0.2\pi, 0.2\pi]$ and $\gamma \in [-\pi, \pi]$ with the step being 0.01π. The best $\mathbf{P}_{3D}$ will be selected from a set of discrete estimations and the corresponding $\hat{A}$ will be used as initial rotation. If more precise results are desired, we can expand the range or shrink the step size, noting that more computation time will be required.

Sample single image results have been shown in Figures 1 and 2. More results in Figure 9 shows unusual poses where the animals stand on two feet. In the supplementary material, we include animations showing more novel viewing angles of the estimated 3D skeletons. Table 1 tabulates the reprojection error of different animal species, where the skeletons are normalized to the range of $[-6, 6]$ during optimization. We observe that the error is higher than average when the animal is in unusual pose, such as a standing panda walking to catch food. Also, a higher error is observed for animals such as the lion cub which is too furry to accurately determine the position of its joints.

Species	horse	panda	raccoon	cub	bear	alpaca
Error	0.133	0.320	0.277	0.284	0.196	0.244

Table 1. Mean reprojection error or mean 2D per joint position error. We have 15 horse videos and 1 video for the rest of each animal species. Each clip contains 30 frames.

5.2. Videos

Species ($\times 10^{-2}$)	horse	panda	raccoon	cub	bear	alpaca
Stability (before)	0.93	2.8	0.72	4.2	2.7	3.1
Stability (after)	0.47	0.79	0.46	0.85	0.67	1.2

Table 2. Improved temporal stability of the 3D animal skeletons before and after temporal module. Same video clips as in Table 1.

This section evaluates the results of our proposed temporal module. The qualitative comparison in Figure 10 shows that our temporal network can successfully predict the occluded joints while the 2D-to-3D network alone fails to do so. An extensive set of video results in GIF format can be found in the supplementary material.

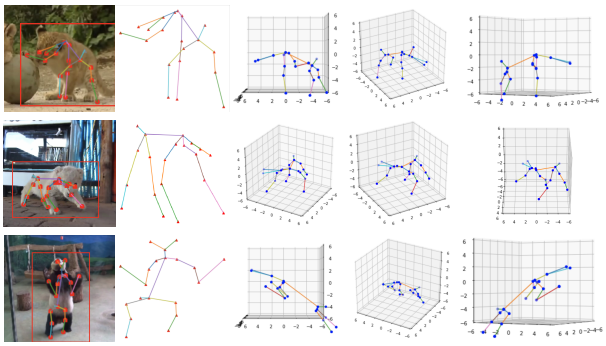

Figure 9. More results of our single video module as a continuation of Figure 2.

We compute the temporal stability before and after the temporal module to validate its efficacy. Temporal stability is defined as the mean second-order motion (acceleration) for each 3D joint in a video sequence. Table 2 shows the improved temporal stability when temporal coherence is considered by our temporal module.

As existing animal dataset [8] or self-labeled data do not provide ground-truth 3D joint positions, we can resort to report the mean 3D joint-wise error on our synthetic dataset. Our validation dataset contains 15K synthetic videos and each video contains 100 frames. The mean 3D joint-wise error for the validation dataset is 1.032 if only the single image module is used, while the error decreases to 0.837 after the temporal module thus demonstrating its effectiveness.

5.3. Ablation Study

5.3.1 Different Architectures for 2D-to-3D Network

In previous works such as [11] the most common architecture for 2D-to-3D estimation is similar to ours in section 3.3 (Figure 5), but with 2 residual blocks instead of 3. Table 3 shows the diminishing gain when the number of layers in-

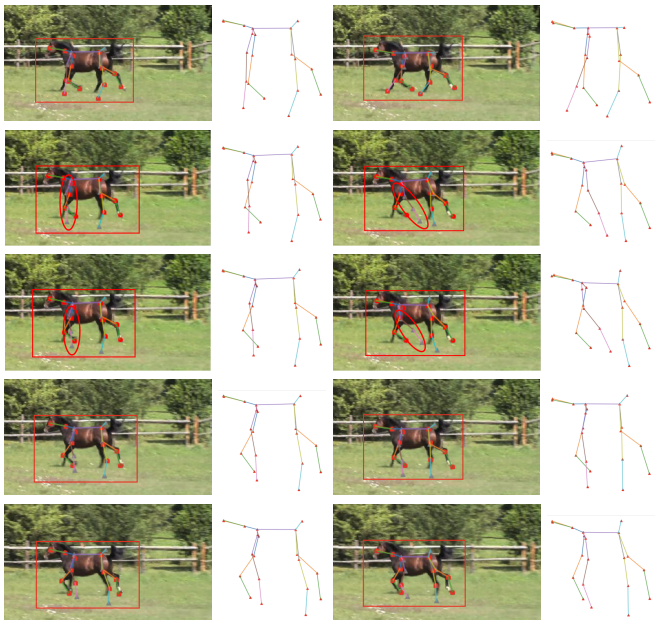

Figure 10. Left two columns show the results refined by our temporal network. Right two columns show the raw output from the 2D-to-3D network. It is apparent that the raw results falsely predict the leg position under self-occlusion, whereas our temporal network can correct the mistakes using information from neighboring frames. Please view supplementary videos.

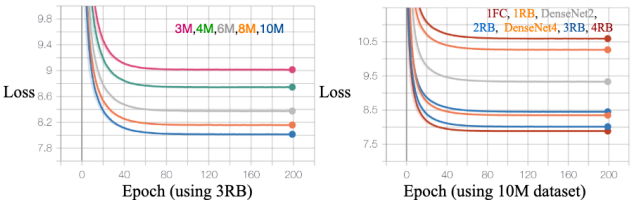

Figure 11. Loss versus different setups. Left shows the effect of size of the dataset and right of different network modules.

creases, showing that while the result of using 3 residual blocks is better than that of using 2 residual blocks, the improvement from 2 blocks to 3 blocks is much smaller than that from 1 block to 2 blocks. We finally choose 3 residual blocks to keep the module simple while maintaining a reasonable accuracy. In addition, we test the Densenet [18] architecture in the supplementary material.

Module	Loss
1 Fully Connected Block (Baseline)	10.590
1 Residual Blocks	$10.270_{\downarrow 0.320}$
2 Residual Blocks	$8.452_{\downarrow 1.818}$
3 Residual Blocks (final choice)	$8.015_{\downarrow 0.437}$
4 Residual Blocks	$7.886_{\downarrow 0.129}$

Table 3. Losses on different number of residual blocks in 2D-to-3D network.

5.3.2 Different dataset sizes for 2D-to-3D Network

As the training set is synthesized, it is easier to obtain than those acquired in the real world. Thus it is possible to try different sizes in order to find a balance between accuracy and training time. Table 4 shows the pertinent losses on using datasets of different size. Figure 11 shows that a dataset larger than 7 millions is in general sufficient to provide an accurate network while maintaining a reasonable training time. In our experiments, we use 10 millions to achieve the best prediction despite longer running times.

5.3.3 Convolution layers in temporal network

The number of convolution layers used in our temporal network can influence prediction quality. For example, if we use more than necessary, the network may "see" more frames than necessary at one time which may lead to false prediction as shown in Figure 12. Observe that the distance between the two front legs in the right two columns of the figure is much smaller than it should be. Because the legs move in a wide range within a longer sequence, the predicted results tend to average the input frames and cannot reflect the true movement. Table 5 shows the mean 2D mean per joint position errors for 2 and 3 convolutional layers for different animals (same normalization as in section 5.1) and

Size	3M	4M	6M	8M	**10M**
Loss	9.016	8.745	8.374	8.158	8.015

Table 4. Losses using different size of the synthetic dataset. 10M is our final choice.

we see that in all but one cases, the error increases from a 2-layer setup to 3-layer setup. Thus we choose the 2-layer setup in our experiments.

Species	horse	panda	raccoon	cub	bear	alpaca
Error (2 layers)	0.302	0.927	0.363	0.442	0.420	0.613
Error (3 layers)	0.333	0.970	0.350	0.482	0.459	0.669

Table 5. Mean reprojection error for different convolutional layer setups. Same video clips used as in Table 1.

Interestingly, when we compare the reprojection errors in Table 1 for single images and Table 5 for videos, the image errors are *smaller* than the video errors which is counter-intuitive. In hindsight, this is reasonable, because reprojection errors are computed based on non-occluded points, which will be affected once we consider the temporal coherence, and the improvement for occluded points can not be reflected in the reprojection error metric.

5.4. Limitation of the Framework

There are few cases where our framework does not perform well. First, if the animal is very close to the camera, the highly perspective image will break our assumption of

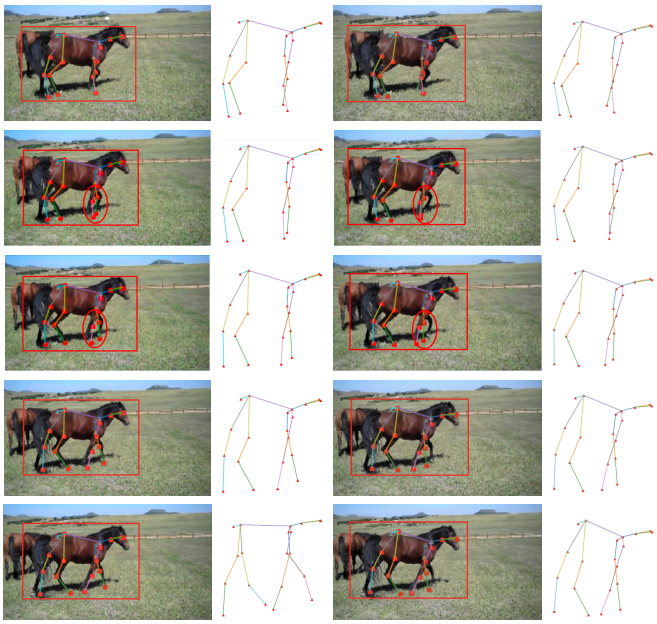

Figure 12. The left two columns show the results with 2 convolutional layers (9 frames concatenated each time; our setup) and the right two columns show the results with 3 convolutional layers (27 frames concatenated each time).

symmetry as the near limb appears longer than the distant limb. Second, if the animal is very furry or under severe self-occlusion, the 2D landmark detection can not provide us with sufficient and accurate joint landmarks as input to predict the pertinent 3D skeleton.

6. Conclusion

In this paper, we present a novel parametric approach for generating and estimating the 3D skeleton of quadrupedal animals, and provide a large-scale synthetic dataset for public use with two deep neural networks as contributions. Extensive experiments conducted on existing animal dataset and self-labeled videos have demonstrated that our method is applicable to animals with different species, dimensions and poses. We believe that this work will advance the development of more realistic animal animation in computer graphics, and contribute to animal behavioral study for biological and behavioral scientists.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1446–1455, 2015.

[3] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5167–5176, 2018.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition(CVPR)*, pages 3686–3693, 2014.

[5] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4733–4742, 2016.

[6] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Personalizing human video pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3063–3072, 2016.

[7] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 7035–7043, 2017.

[8] Luca Del Pero, Susanna Ricco, Rahul Sukthankar, and Vittorio Ferrari. Behavior discovery and alignment of articulated object classes from unstructured video. *International Journal of Computer Vision*, 121(2):303–325, 2017.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[10] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Dylan Drover, Rohith MV, Ching-Hang Chen, Amit Agrawal, Ambrish Tyagi, and Cong Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[12] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[13] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[14] Wolfgang Graf, Catherine de Waele, and Pierre Paul Vidal. Functional anatomy of the head-neck movement system of quadrupedal and bipedal mammals. *Journal of Anatomy*, 186(Pt 1):55, 1995.

[15] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[16] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7297–7306, 2018.

[17] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[20] 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, 2013.

[21] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5, 2010.

[22] Hamza Khan, Roy Featherstone, Darwin G Caldwell, and Claudio Semini. Bio-inspired knee joint mechanism for a

hydraulic quadruped robot. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 325–331. IEEE, 2015.

[23] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 6050–6059, 2017.

[24] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[26] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[27] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Technical report, Nature Publishing Group, 2018.

[28] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017.

[29] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML)*, pages 807–814, 2010.

[30] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie W Mathis. Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature protocols*, 14(7):2152–2176, 2019.

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 483–499. Springer, 2016.

[32] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 7753–7762, 2019.

[33] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3674–3681, 2013.

[34] Daniel Schmitt. Mediolateral reaction forces and forelimb anatomy in quadrupedal primates: implications for interpreting locomotor behavior in fossil primates. *Journal of Human Evolution*, 44(1):47–58, 2003.

[35] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4, 2010.

[36] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[37] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[38] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, 2015.

[39] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660, 2014.

[40] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access*, 6:1155–1166, 2017.

[41] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 109–117, 2017.

[42] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4732, 2016.

[44] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[45] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.

[46] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 398–407, 2017.