Normalized Human Pose Features for Human Action Video Alignment
Supplementary material

A. Illustration of Viewpoint Variation Problem

Figure 1 illustrates the viewpoint variation problem of using joint position-based pose representation for feature extraction with a toy example. When a vector $\overrightarrow{AB}$ in the world coordinate is recorded by the two frontal-view cameras in the Human3.6M dataset, the vectors $\overrightarrow{A'B'}$ and $\overrightarrow{A''B''}$ differ vastly between the two viewpoints. In practice, points A and B could represent the root joint and another joint of a subject, respectively. When recording the same pose in the world coordinate, the joint positions relative to a root joint would appear vastly different among different cameras. Without extra modelings, it would be difficult for neural networks or other feature extractors to capture such variations.

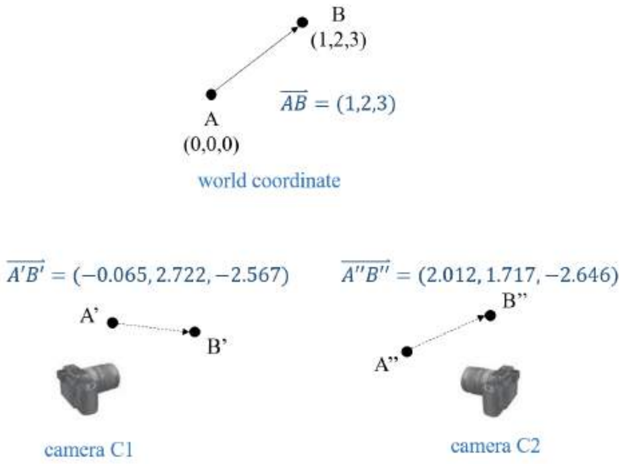

B
(1,2,3)
$\overrightarrow{AB}$ = (1,2,3)
A
(0,0,0)
world coordinate
$\overrightarrow{A'B'}$ = (−0.065, 2.722, −2.567)
A'
B'
camera C1
$\overrightarrow{A''B''}$ = (2.012, 1.717, −2.646)
B"
A"
camera C2

Figure 1. A toy example illustrating the viewpoint variant problem of 3D human poses.

B. Ablation Study on Loss Design

To show the impact of individual losses, we have done an ablation study. See the results in Table 1. It can be seen that when only $\mathcal{L}_{recon}$ is used (1^{st} row) E_Q is still able to converge, but when only $\mathcal{L}_{cycle}$ is used (2^{nd} row), E_Q is hard to generalize on the test set. This indicates that without explicit reconstruction of 3D poses, E_Q can hardly capture the proximity of the poses w.r.t joint rotations. With both losses (from 3^{rd} row and onwards), E_Q is constrained to implicitly pair a pose from a video frame and the projected condition pose, since the two 2D poses are mapped to the same set of joint rotations.

Loss	MPJPE ↓	Δ_{MPJPE} ↓
$\mathcal{L}_{recon}$	58.96	60.97
$\mathcal{L}_{cycle}$	324.97	355.39
$\mathcal{L}_{recon} + \varphi\mathcal{L}_{cycle}$	56.90	59.14
$\mathcal{L}_{recon} + \varphi\mathcal{L}_{cycle} + \beta\mathcal{L}_{jrc}$	53.96	54.23
$\mathcal{L}$ (Ours)	**52.61**	**53.35**

Table 1. Ablation study on the normalization losses. (unit: mm)

C. Implementation Details

The proposed pose normalization and embedding networks are implemented in PyTorch. In data pre-processing, the input 2D poses to E_Q are centered by subtracting the root joint, scaled by frame resolution and then normalized to [-1,1]; the ground-truth 3D poses are root-centered but not scaled, since FK does not contain learnable parameters and a projection is required in the training pipeline. The parameters of the source skeleton s_t are mainly limb lengths, which are computed as the Euclidean distances between physically-connected joints. An alternative way to obtain s_t is to retrieve the bone lengths from the Human3.6M dataset meta-data files. While these two ways result in the same skeletons, directly computing bone lengths from 3D poses allows easy generalization to our bone-augmented dataset, as well as other potential 3D human pose datasets.

The two networks E_Q and E_P were trained in different stages, i.e., we first trained the network for pose normalization and then froze the parameters and trained the pose embedding network using the output of the former. In the learning of pose normalization, $\varphi = 0.6$, $\beta = 1$, and $\lambda = 0.1$ in our experiments. In the learning of pose embeddings, the positive range was set as $[0, 0.2]$; the negative range was set as $[0.8, 1.0]$ at the beginning of the training as easy negative, and changed into the range $[0.5, 0.75]$ for harder negative samples. We adopted $d = 64$ as the dimension of pose features.

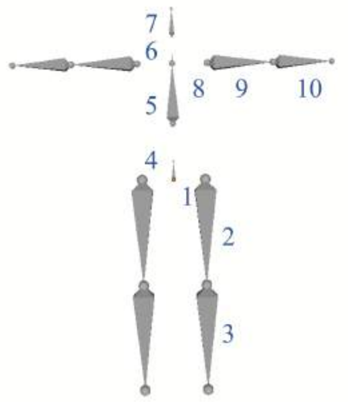
The details of the architecture of the pose normalization network E_Q in Section 3.2 is shown in Table 2. The architecture of the pose embedding network E_P in Section 3.3 is the same as that in the paper [2], except for the only difference that the output dimension is changed into the dimension of pose feature, instead of the number of joint position parameters.

#	Layer	Dim	Kernel	Stride	Padding
1	Conv1d-Norm	(34,1024)	1	1	0
2	Conv1d-Norm	(1024,1024)	5	1	0
3	Conv1d-Norm	(1024,1024)	3	1	0
4	Conv1d-Norm	(1024,1024)	1	1	0
5	Conv1d-Norm	(1024,1024)	5	1	0
6	Conv1d-Norm	(1024,1024)	3	1	0
7	Conv1d-Norm	(1024,1024)	1	1	0
8	Conv1d	(1024,68)	1	1	0

Table 2. Network architecture for E_Q.

D. Parameters of Condition Skeleton

Figure 2 shows the definition of 10 bone segments of the condition skeleton and their corresponding lengths. The condition skeleton is bilaterally symmetric in bone lengths. Their lengths are defined according to the average bone lengths in the training set of Human3.6M. During training, these bone lengths are used in the FK in the cycle reconstruction branch (Figure 3(c)) to compute the positions of the 17 joints in the condition skeleton.



Index	Length
1	135
2	450
3	445
4	238
5	255
6	116
7	115
8	149
9	280
10	250

Figure 2. The joints and pre-defined bone lengths of the condition skeleton. (unit:mm)

E. Dense Correspondence Dataset

In this section we provide more details on our synthetic dense correspondence dataset based on the Human 3.6M dataset. Of the four synchronized videos recorded from four viewpoints for each action, we used the two in frontal views, as shown in an example in Figure 3. The two corresponding frames share the same subject pose, but they differ in 2D joint positions and scales of 3D skeletons in camera coordinates.

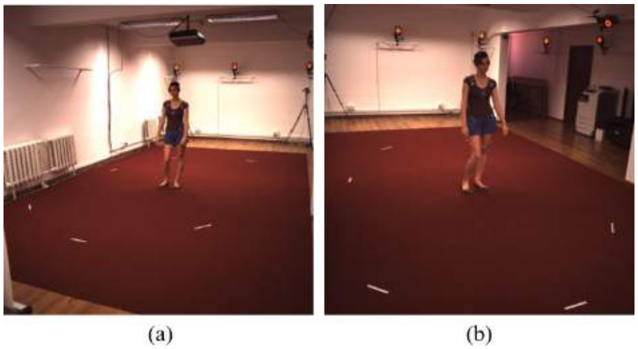


Figure 3. An example of frames in the in two frontal-view videos used for building the correspondence dataset.

For each pair of the videos, we produced the difference in length of source and target videos by re-sampling the frames, and then forming new correspondences. For example, in Figure 4(b), in the source sequence (gray dots), the indexes of retained frames are (0,2,3,6,8,9); and those in the target sequence are (1,3,4,5,7,10,11). Since the original video frames are temporally aligned (Figure 4(a)), where correspondent frames have the same frame index, the frame indexes can be used as a feature representing similarity. Applying DTW on frame indexes yields the ground-truth correspondence:

$$C = \{(0,0),(1,1),(2,1),(2,2),(3,3),(4,4),(5,5),(5,6)\},$$

as illustrated in Figure 4(c).

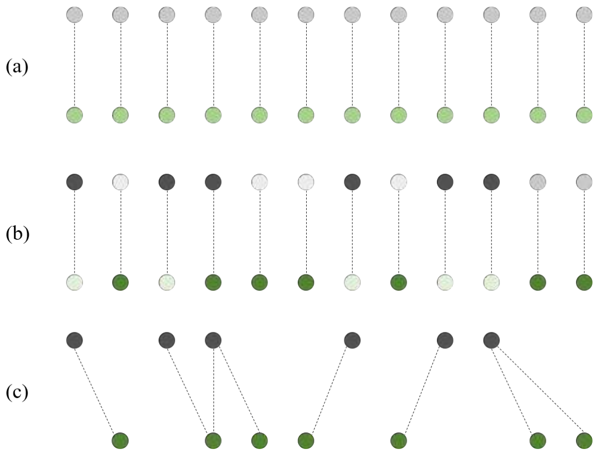


Figure 4. This figure illustrates the construction of synthesized correspondence dataset. (a) The originally aligned source (represented by gray dots) and target (green dots) videos; (b) the video frames are re-sampled by randomly dropping certain frames; (c) the new correspondence is determined by the DTW algorithm on frame indexes.

F. Embedding Space Visualization

To track the poses in the embedding space, we visualize the trajectories of action videos in Figure 5 (the transparency encodes time). The loop-like shapes in the dotted frames in the weight-lifting action (Figure 5(a)) correspond to the half-squat and rise up in the action. In the tennis serve example (Figure 5(b)), the starting and ending poses are more similar, making the two ends of the trajectories closer in the embedding space. The action of jumping-jacks is highly periodical, and thus the trajectory of continuous jumping-jacks also appears repetitive (Figure 5(c)).

G. More Visualization Results

In this section we demonstrate more pose normalization results on the Penn Action dataset, as shown in Figure 6. The Penn Action dataset contains challenging poses from sport actions. The dataset provides video frames and ground-truth 2D poses, but without ground-truth 3D poses. Since the 2D poses in the Penn Action Dataset only contains 13 joints, we adopted OpenPose [1] to detect 2D poses from the video frames for the compatibility of joint numbers. We compare the results with Procrustes aligned 3D poses by the same 3D pose estimation method [2] as in Section 5.2. For 3D poses computed by the joint position estimation method, it is difficult to unify their global orientations by aligning them with a pre-defined T-pose (Figure 6(e)), especially when the poses are complex and when the ground-truth 3D poses are unknown. In contrast, the normalization of global orientation can be easily achieved with joint rotations (Figure 6(g)). Our pose normalization method can generalize to poses that are unseen during training, such as baseball pitches.

H. Human Pose Retrieval

To evaluate the generalization ability of our pose features to other pose similarity tasks, we conducted a pose retrieval task on the MPI-INF-3DHP (3DHP) dataset [3], as shown in Figure 7. We did not fine-tune our models and performed the task directly on the 3DHP testset. For a query frame in a video, we retrieved the top-3 frames with similar poses in all the other videos using the L2 distance of our pose features. Even though the subjects were performing different actions, the retrieved poses resemble the query poses with respect to the pose characteristics, e.g. bending over or arm raised.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 3, 5

[2] 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*, pages 2640–2649, 2017. 1, 3, 5

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

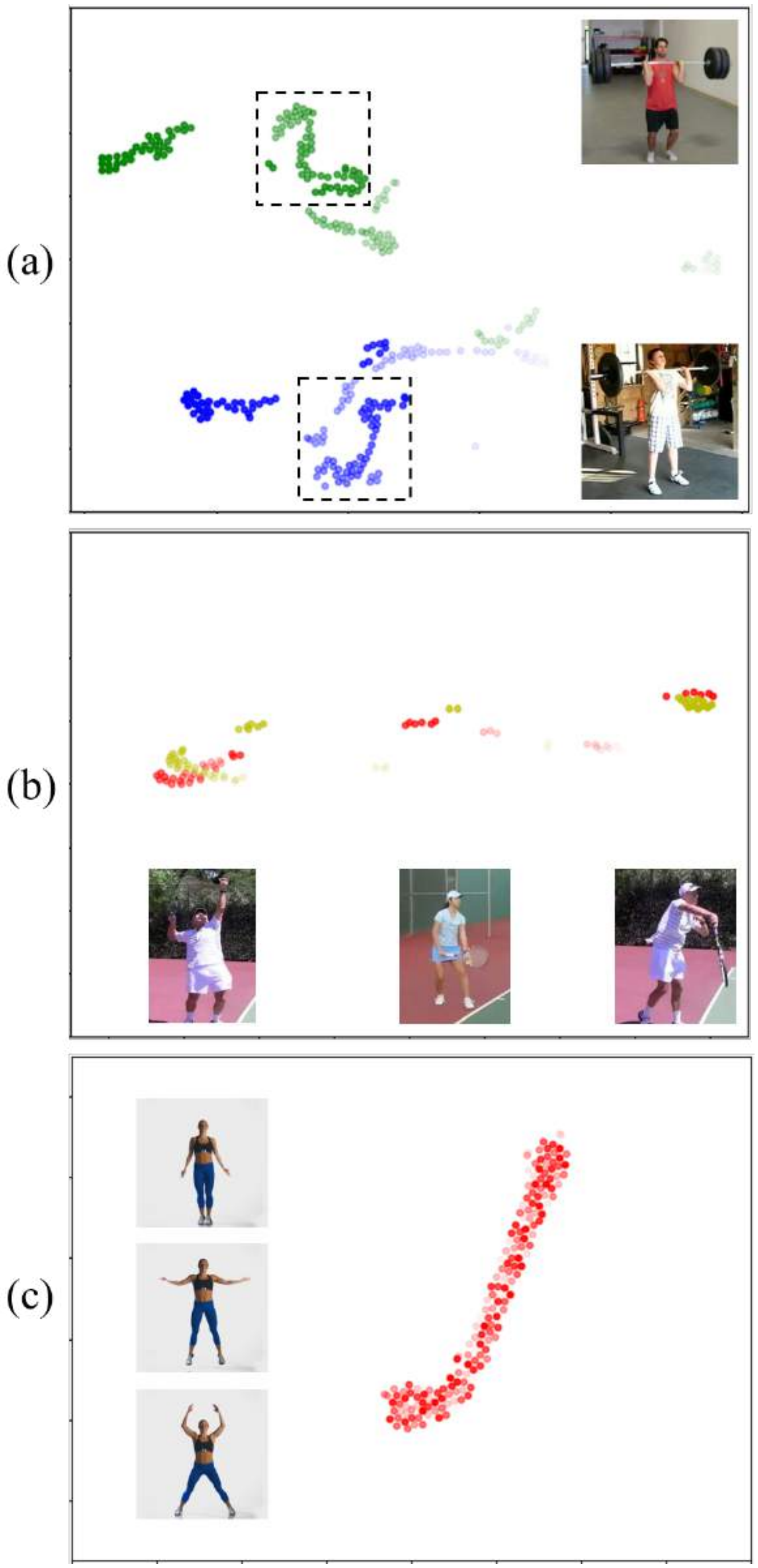


Figure 5. A visualization of action trajectories in the embedding space.

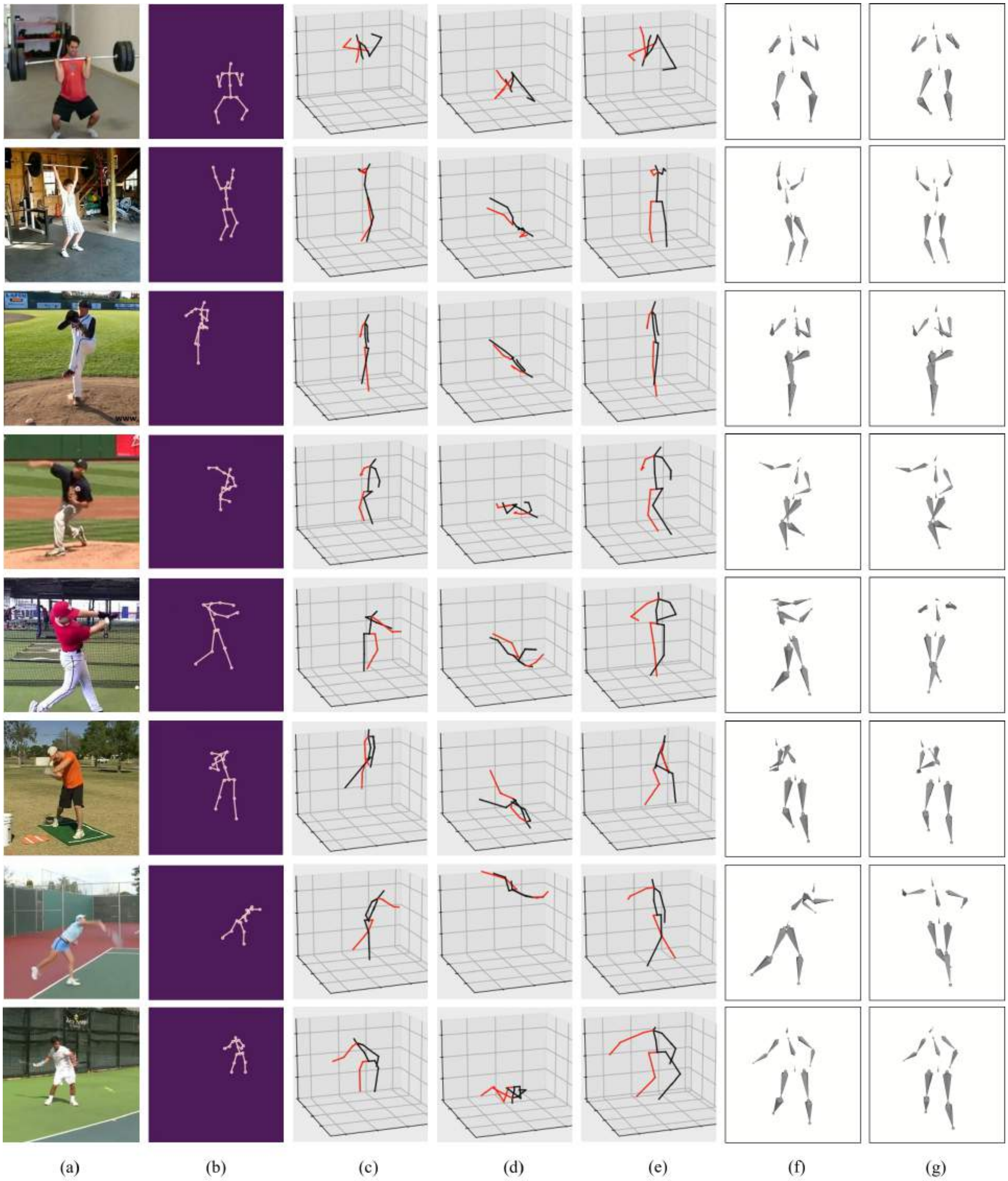


Figure 6. Visualization of pose normalization results. (a) The video frames from the Penn Action dataset; (b) the 2D poses detected from video frames by OpenPose [1]; (c) estimated 3D poses by Martinez [2] in world coordinates; (d) 3D joint positions in camera coordinates; (e) unify 3D poses in camera coordinates by Procrustes alignment with a pre-defined T-pose; (f) 3D condition skeleton poses by our method; (g) our normalized 3D poses under a unified global orientation.

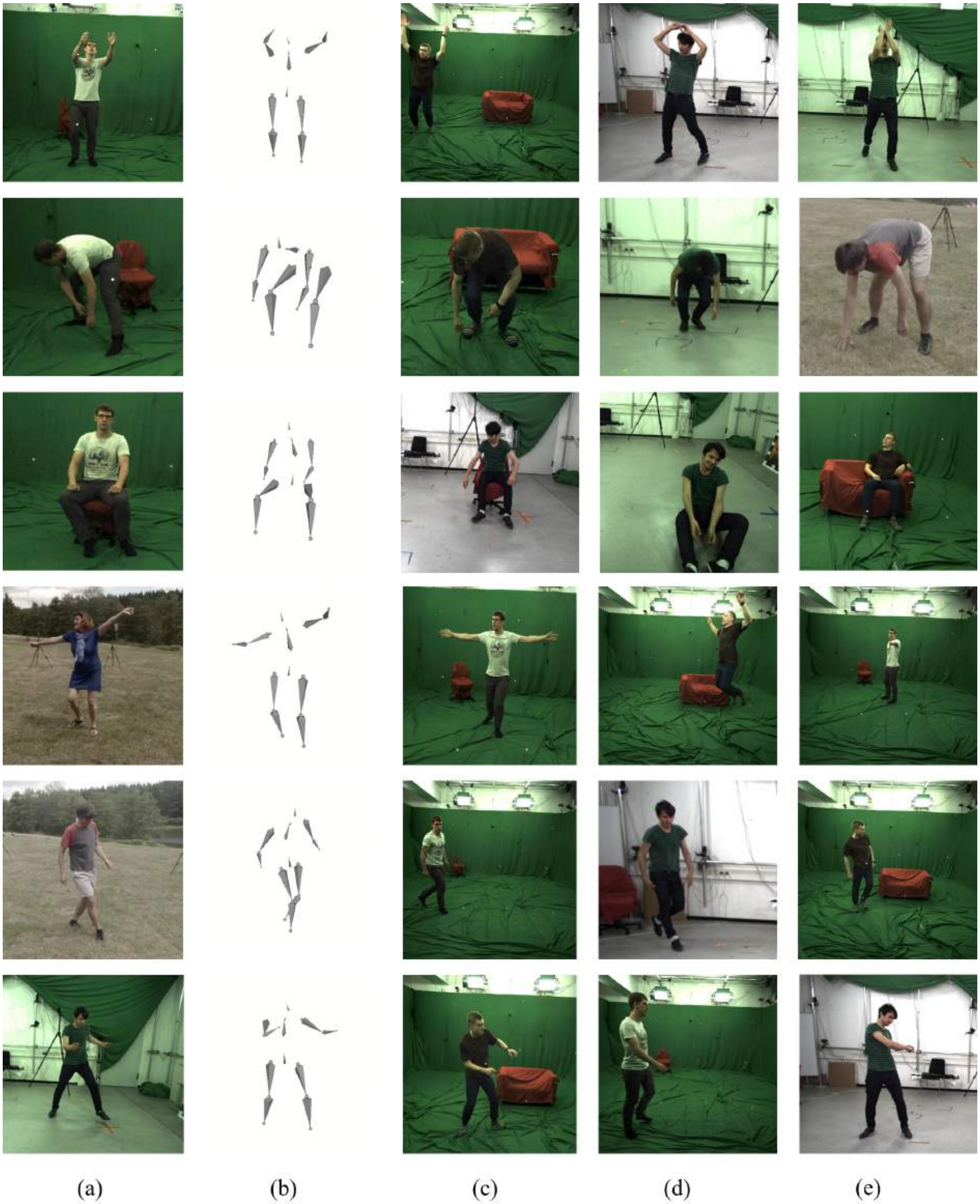


Figure 7. Visualization of human pose retrieval on the 3DHP test set. (a) The video frame of query poses; (b) the normalized query poses (with global orientations); (c)-(e) top-3 retrieved poses using our pose features.