—Supplementary Materials— DeciWatch: A Simple Baseline for 10× Efficient 2D and 3D Pose Estimation

Ailing Zeng[1], Xuan Ju[1], Lei Yang[2], Ruiyuan Gao[1], Xizhou Zhu[2], Bo Dai[3], and Qiang Xu[1]

[1]The Chinese University of Hong Kong, [2]Sensetime Group Limited,
[3]Shanghai AI Laboratory
{alzeng, qxu}@cse.cuhk.edu.hk

In Sec. 1, we present dataset descriptions. Next, we present results of efficient labeling in Sec. 2 and the generalization ability of *DeciWatch* in Sec. 4. Then, we show more ablation studies on different sampling ratios, model designs of *DenoiseNet* and *RecoverNet*, and hyper-parameters in Sec. 5. Moreover, we show qualitative comparison results in Sec. 6 to demonstrate why *DeciWatch* works. Last, in Sec. 7, we discuss some failure cases in this method to motivate further research.

1 Dataset Descriptions

– **Sub-JHMDB** JHMDB[3] is a video-based dataset for 2D human pose estimation. For a fair comparison, we only conduct our experiments on a subset of JHMDB called sub-JHMDB. It contains 316 videos and the average duration is 35 frames. For each frame, it provides 15 annotated body keypoints. We use the bounding box calculated from the puppet mask provided by [9]. Following the settings [17, 1], we mix 3 original splitting schemes for training and testing together in 2D pose estimation experiments.
– **Human3.6M** [2] Human3.6M is a large-scale indoor video dataset with 15 actions from 4 camera viewpoints. It has 3.6 million frames and a frame rate of 50 fps. 3D human joint positions are captured accurately from a high-speed motion capture system. Following previous works [15, 11, 12, 16], we use the standard protocol with 5 actors (S1, S5, S6, S7, S8) as the training set and another 2 actors (S9, S11) as the testing set.
– **3DPW** [10] 3DPW is a challenging in-the-wild dataset consisting of more than 51k frames with accurate 3D poses and shapes annotation. The sequences are 30fps. This dataset is usually used to validate the performance of model-based body recovery methods [5, 7, 4, 6].
– **AIST++** [8] is a challenging dataset with diverse and fast-moving dances that comes from the AIST Dance Video DB [13]. It contains 3D human motion annotations of 1, 408 video sequences at 60 fps, which is 10.1M frames in total. The 3D human keypoint annotations and SMPL parameters it provides cover 30 different actors in 9 views. We follow the original settings to split the training and testing sets based on actors and actions.

2 An application: Efficient Pose Labeling in Videos

Due to the efficiency and smoothness of the pose sequences recovered by *DeciWatch*, reducing the need for dense labeling could be a potential application. We verify the effectiveness of this application on the Human3.6M and AIST++ dataset by directly inputting the sparse ground-truth 3D positions into the *RecoverNet* of *DeciWatch*. In Table 1, we compare *DeciWatch* with the most used spline interpolation, linear interpolation, and quadratic interpolation. Our method has a slower error growth as the interval N gets larger. To be specific, it is possible to label one frame every 10 frames with only 2.89mm position errors in slow movement videos (e.g., in Human3.6M [2]) and label one frame every 5 frames with only 4.03mm position errors in fast-moving videos (e.g., in AIST++ [8]). This application can improve annotation efficiency by more than 10×.

Table 1. Comparison of *MPJPE* on efficient pose labeling that labels one frame in every N frames on Human3.6M [2] and AIST++ [8] dataset.

	Human3.6M					**AIST++**				
Interval N	2	5	10	15	20	2	5	10	15	20
Linear	2.21	6.55	10.81	24.15	35.20	7.21	21.31	27.72	73.69	99.04
Quadratic	1.26	4.31	10.05	17.22	22.85	2.04	8.33	23.59	43.13	61.16
Cubic-Spline	**0.18**	**0.99**	5.36	18.42	29.21	0.89	5.12	18.31	45.32	77.39
DeciWatch	0.25	1.33	**2.89**	**6.21**	**10.59**	**0.83**	**4.03**	**11.25**	**20.12**	**41.25**

3 Additional Evaluation Metrics for 2D Pose Estimation

As is shown in Tab. 1 in the main paper, the results of *DeciWatch* have achieved nearly 99% accuracy on PCK@0.2. However, qualitative visualization shows that an awful lot of errors still exist in the recovery results. We attribute it to the fact that PCK@0.2 is quite loose for accuracy measurement, which only requires the detected keypoints to be within 20% of the bounding box size under pixel level. As a result, we use two additional evaluation metrics, PCK@0.1, and PCK@0.05, for better localization evaluation. More specifically, PCK@0.1 and PCK@0.05 restrict the matching threshold to 10% and 5% of the bounding box size. Tab. 2 shows the results of *DeciWatch* and SimplePose[14] on these three metrics. In future work, we recommend using PCK@0.05 as the main metrics for 2D pose estimation.

Table 2. Comparison of *DeciWatch* and SimplePose[14] on PCK@0.2, PCK@0.1, and PCK@0.05. In future work, we recommend using PCK@0.05 as the main metrics for 2D pose estimation.

Sub-JHMDB dataset - 2D Pose Estimation				
Sampling Ratio	Evaluation Metric	PCK@0.2 ↑	PCK@0.1 ↑	PCK@0.05 ↑
20%	SimplePose	93.92%	81.25%	56.88%
	DeciWatch	**99.11%**	**95.43%**	**82.66%**
10%	SimplePose	93.94%	81.61%	57.30%
	DeciWatch	**98.75%**	**94.05%**	**79.44%**
5%	SimplePose	92.38%	82.79%	58.95%
	DeciWatch	**97.50%**	**91.76%**	**73.02%**

4 Generalization Ability

DeciWatch learns the patterns of noisy human motions since motion distribution could be overlapped among some datasets, making it has potential generalization ability. We further test *DeciWatch* trained on 3DPW-PARE across various backbones and datasets in Tab. 3, where *DeciWatch* still achieves competitive pose estimation results with 10x efficiency. We attribute it to the fact that *DeciWatch* effectively learns the continuity of motions, which is applicable for different sorts of motions.

Table 3. Cross-backbone and cross-dataset results from *DeciWatch* checkpoints trained on 3DPW-PARE.

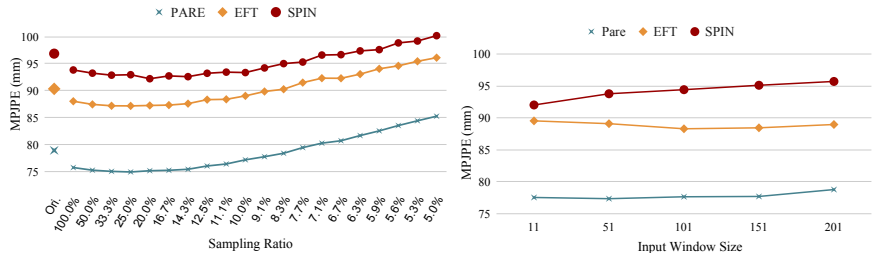
Dataset/Estimator		MPJPE↓	Accel.↓
3DPW/EFT	Estimator (100%)	90.3	32.8
	DeciWatch (**10%**)	**87.2**$\downarrow_{3.1(3.4\%)}$	**7.2**$\downarrow_{25.6(77.9\%)}$
3DPW/SPIN	Estimator (100%)	**96.9**	34.6
	DeciWatch (**10%**)	98.3$\uparrow_{1.4(1.4\%)}$	**7.1**$\downarrow_{27.5(79.5\%)}$
AIST++/SPIN	Estimator (100%)	107.7	33.8
	DeciWatch (**10%**)	**101.8**$\downarrow_{5.9(5.5\%)}$	**6.2**$\downarrow_{27.6(81.7\%)}$

5 Ablation Study

Impact of sampling ratio and input window size. Both input window size and sampling ratio will affect inference efficiency and performance of *DeciWatch*. With the same input window size, the lower the sampling ratio is, the more efficient the inference process will be. We present the comparison of original pose estimator (Ori.)and *DeciWatch* with sampling ratio from 100% to 5% (sampling interval N changes from 1 to 20) in Fig. 1(a). When the sampling ratio is 100%, *DeciWatch* can be regarded as a denoise model. As shown in Fig. 1(a), the changing trends of *MPJPE* are similar for all three estimation methods (PARE [6],

EFT [4], SPIN [7]). Surprisingly, we find that *MPJPEs* first drop before rising, and they are smallest when the sampling ratio is about 20%, with improvements of 4.9%, 3.4%, and 4.9% for PARE, EFT, and SPIN respectively. This gives us a new perspective that in pose estimation, *not every frame has to be watched to achieve better performance*. The reason behind this is the different degrees of noise in estimated poses. It may be harder to eliminate these diverse degrees of errors in all frames than only denoise some of the frames and recover the rest by temporal continuity. Besides, the *MPJPEs* of *DeciWatch* is worse than that of the original pose estimator when the sampling ratio is larger than 8% due to too limited input information.

With the sampling ratio fixed at 10%, we further explore the influence of window size. In Fig. 1(b), we test window sizes from 11 to 201. Results indicate that our framework is robust to different window sizes.



(a) Effect of the sampling ratio (b) Effect of the window size

Fig. 1. Comparing effects of different (a) sampling ratios and (b) window sizes. Sampling interval N is from 1 to 20. We compare *MPJPEs* of the three original (Ori.) pose estimators [6, 4, 7] to our framework on the 3DPW dataset.

To serve for future research, we report results, including *MPJPEs* and *Accels*, of 3D pose and body estimation on 3DPW, Human3.6M, and AIST++ datasets in Table 4. All results show similar trends in the change of precision (*MPJPEs*) and smoothness (*Accels*). In addition, *DeciWatch* utilizes the natural smoothness of human motions to recover the detected poses. As a result, the *Accels* decreases steadily when the interval N increases, indicating *DeciWatch* can enhance the smoothness of the existing backbone methods.

Analyses on the phenomenon: fewer samples with better performance. When the inputs of *DeciWatch* are ground-truth poses, the performance deteriorates with decreased sampling ratio (see Table 1 above). However, in practice, the inputs of *DeciWatch* are noisy detected poses, and some of them have high errors (e.g., due to occlusion). Consequently, considering the detected poses' errors, two factors affect the recovered poses.

1) On the one hand, not considering/aggregating the highly noisy poses can improve performance by reducing the impact of noisy poses on both *DenoiseNet*

Table 4. Results of original (Ori.) estimators [6, 4, 7, 11] and *DeciWatch* under different sampling ratios. Ori. is the watch-every-frame pose estimator. Sampling interval N is set from 1 to 20. The best results are in bold.

Metrics/N	Ori.	1	3	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PARE [6] Backbone on 3DPW dataset																			
MPJPE	78.9	75.7	**75.0**	75.1	75.2	75.4	76.0	76.4	77.2	77.7	78.4	79.4	80.3	80.7	81.7	82.5	83.5	84.4	85.3
Accel	25.7	25.2	9.2	7.7	7.4	7.2	7.1	7.0	6.9	6.9	6.8	6.8	6.7	6.7	6.7	6.7	**6.6**	**6.6**	**6.6**
EFT [4] Backbone on 3DPW dataset																			
MPJPE	90.3	88.0	**87.2**	**87.2**	87.3	87.6	88.3	88.4	89.0	89.8	90.3	91.5	92.3	92.3	93.1	94.0	94.6	95.4	96.1
Accel	32.8	32.7	10.2	8.0	7.5	7.3	7.1	6.9	6.8	6.8	6.7	6.7	6.6	6.6	6.5	6.5	6.5	**6.4**	**6.4**
SPIN [7] Backbone on 3DPW dataset																			
MPJPE	96.9	93.8	92.9	**92.2**	92.7	92.6	93.2	93.4	93.3	94.2	95.0	95.3	96.6	96.7	97.4	97.6	98.8	99.2	100.2
Accel	34.6	33.5	10.5	8.2	7.7	7.5	7.3	7.2	7.1	7.0	6.9	6.9	6.9	6.9	**6.8**	6.9	**6.8**	**6.8**	**6.8**
FCN [11] Backbone on Human3.6M dataset																			
MPJPE	54.6	53.3	53.0	52.8	52.6	**52.3**	52.5	52.6	52.8	53.0	53.2	53.2	53.4	53.5	53.9	53.8	54.0	54.2	54.4
Accel	19.2	15.4	3.1	2.0	1.8	1.6	1.6	1.5	1.5	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**
SPIN [7] Backbone on AIST++ dataset																			
MPJPE	107.7	67.2	**66.6**	67.6	68.4	69.7	71.2	71.6	71.3	76.1	77.1	79.0	80.2	82.3	84.3	85.2	87.0	88.9	90.8
Accel	33.8	7.6	7.6	6.6	6.3	6.1	6.0	5.9	5.7	5.7	5.6	5.6	5.5	5.5	5.5	5.5	5.4	**5.3**	**5.3**

and *RecoverNet*. 2) On the other hand, dropping too many frames would lead to performance degradation due to information insufficiency.

Generally speaking, when the sampling ratio is high (e.g., >20%), we could easily recover intermediate poses thanks to the continuity of motions. And for those dropped intermediate poses, the denoised and recovered poses via *DeciWatch* obtain lower error compared to their original noisy estimation. Consequently, the overall MPJPEs would drop with the increase of intervals (i.e., the decrease of sampling ratio) in the beginning. However, when the sampling ratio becomes too low (e.g., <5%), the highly sparse poses do not provide sufficient information for motion recovery, and the MPJPEs would go up under such circumstances. In other words, there would be a "sweet spot" for the sampling ratio with the minimum MPJPEs. This phenomenon is also present in the traditional interpolation method, where *MPJPE* first drops (from 107.7mm to 105.8mm) before rising.

Table 5. Comparison results with different denoise network designs on 3DPW dataset with the state-of-the-art pose estimator Pare [6] (*MPJPE* is 78.9mm).

Metrics	No *DenoiseNet*	TCNs [12]	MLPs [16]	Ours
MPJPE	79.8	80.5	79.5	**77.2**

Study on different denoise networks. As a baseline framework, we try to explore the performance of different network designs of the two subnets, *DenoiseNet* and *RecoverNet*. In the second step, we use *DenoiseNet* with Transformer architecture to relieve noises from single-frame estimators. We first remove this network to validate its effectiveness. Table 5 shows a 2.1mm reduction

of *MPJPE* without *DenoiseNet*, indicating this step is essential to the recovery of more precise poses. Then, we try to simply replace the Transformer with TCNs [12], with zero paddings to make the input and output length the same, and MLPs [16] along temporal axes. Results show these models are incapable of handling the discrete diverse noises, making the final recovery results worse than the original result.

Table 6. Comparison results with different recovery network designs on 3DPW dataset with the state-of-the-art pose estimator PARE [6](*MPJPE* is 78.9mm).

Metrics	Linear	TCNs [12]	TCNs w/MLPs	MLPs [16]	Ours
MPJPE	79.8	172.3	99.5	78.0	**77.2**

Study on different recovery methods. Lastly, we analyze possible designs of the recovery process in the third step. First, we try the simple Linear interpolation, which shows more significant errors compared with the original PARE since it loses the non-linear motion dynamics. Then, we adopt TCNs [12], which have local temporal receptive fields (e.g., 3) in each layer to recover the missing values with the interval N as 10, and it leads to the worst results. After adding MLPs [16] at the last layer to enhance long-term temporal coherence, the error reduces from 172.3mm to 99.5mm (by 42.3% improvement), but the error is still far from satisfactory. MLPs [16] can utilize the continuity of temporal dimension to learn non-linear fitting curves from sampled points. Still, they do not aggregate spatial information, which makes them get a slightly worse result.

Table 7. Comparison results of different loss weight λ on 3DPW dataset with the state-of-the-art pose estimator Pare [6](MPJPE is 78.9mm).

λ	1	2	5	10
MPJPE	78.0	77.6	**77.2**	77.5

Study on different hyper-parameters. We also show the effects of hyper-parameters in *DeciWatch*. λ is used in the loss function to balance the losses between *RecoverNet* and *DenoiseNet*. Results in Table 7 show that *MPJPEs* are robust to diverse loss values. Therefore, we set it to 5 by default. Moreover, we use the same embedding dimension C and block number M in transformer blocks. We show the results of different C in Table 8 and M in Table 9. Fewer parameters, such as $C = 12$ and $M = 1$, will lead to performance degradation. As the model becomes deeper (larger M) and wider (larger C), the performance will meet saturation. By default, we set $C = 64$ and $M = 5$ for all experiments.

Table 8. Comparison results of different embedding dimension C on 3DPW dataset with the state-of-the-art pose estimator Pare [6].

C	12	32	64	128	256
MPJPE	78.0	**77.2**	77.4	77.6	77.4

Table 9. Comparison results of different block number M on 3DPW dataset with the state-of-the-art pose estimator Pare [6].

λ	1	3	5	10
MPJPE	79.3	77.5	**77.2**	77.6

6 Qualitative Results

We demonstrate three typical successful cases of *DeciWatch* to understand why *DeciWatch* uses fewer frames with higher efficiency but gets better performance than existing single-frame methods.

First, cases in Fig. 2 show that *DeciWatch* can improve not only efficiency but also effectiveness on the 3D body recovery task. The estimated body in the yellow boxes are inputs of *DeciWatch*, where the interval N is set to 10. Existing SOTA models, like PARE [6], will fail (illustrated in red boxes) when the frames have heavy body occlusions, human interactions, or poor image quality. Interestingly, *DeciWatch* skips some frames (inputs are in yellow boxes) to avoid the negative effect. Therefore, compared with the watch-every-frame model [6], *DeciWatch* may reduce the effects of unreliable and noisy estimated poses by a temporal recovery scheme to obtain the rest of the results.

What happens if there are mistakes in the visible frames? In Fig. 3, we show the impact of denoising scheme in *DeciWatch* on 2D pose estimation. Given a sliding window of 31 frames, we mainly demonstrate the visible four frames (highlighted in yellow boxes) with their detected 2D poses by SimplePose [14]. We observe that there are left-right flipped keypoint detection in the 1_{th} and 21_{th} frames of Fig. 3(a), which sometimes happens when the input image is the back of the person. In the 31_{th} frame of Fig. 3(d), high errors occur due to heavy self-occlusion. Our method utilizes long temporal effective receptive fields to denoise the noisy input poses and then recover the clean sparse poses to get the final sequence poses, making the output poses smooth and precise in an efficient way.

In addition to being able to do better motion sequence recovery, can DeciWatch *still learn motion prior?* In some cases, even if all visible frames are inaccurate, *DeciWatch* can still recover accurate poses by learning motion prior.

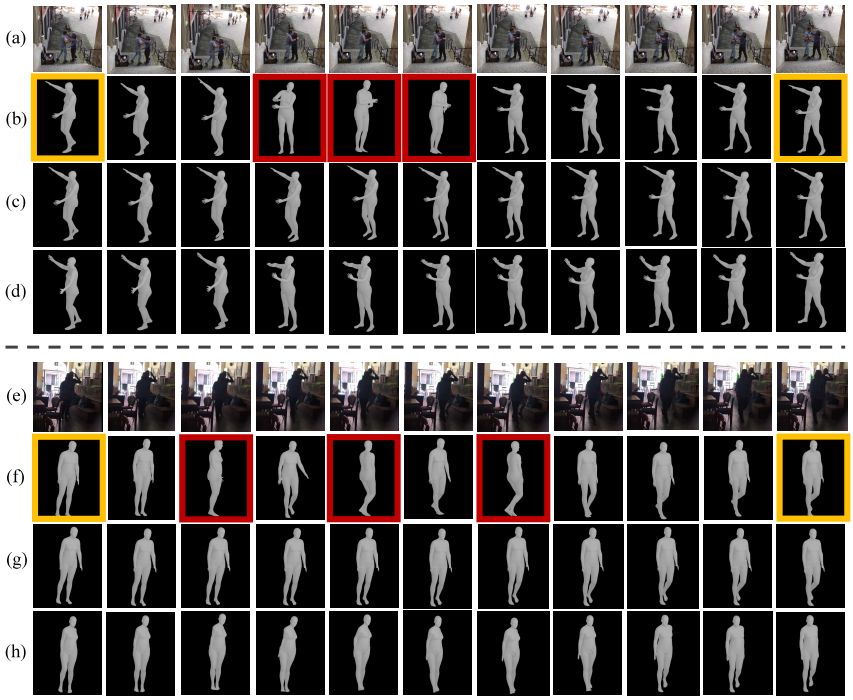


Fig. 2. Visualization results of estimated body recovery from two video sequences with eleven frames in (a) and (e) rows. (b) and (f) are estimated poses from the existing SOTA model PARE [6]. We highlight the input poses of *DeciWatch* in the yellow boxes and the high-error poses in the red boxes. (c) and (g) are output poses of our proposed *DeciWatch*, the sampling ratio is 10% in this framework. (d) and (h) show the ground truth of the corresponding poses.

As shown in Fig. 4, (a) shows the original video frames of AIST++ with an interval of 10, which is all the visible frames in one slide window (a sliding window with the length of 101 has 11 visible frames). (b) is the corresponding SMPL pose detected by SPIN [7]. Large errors occur in the actor's occluded right arm and hand. In Fig. 4(c), *DeciWatch* can successfully correct the errors and outputs smooth poses leveraging dancing action prior and human motion continuity, which are hard for existing single-frame estimators to estimate occluded body parts. (d) shows the ground truth poses of the video frames.

For more visualization of 2D pose estimation, 3D pose estimation as well as body recovery, please refer to our website[1].

[1] Website: https://ailingzeng.site/deciwatch

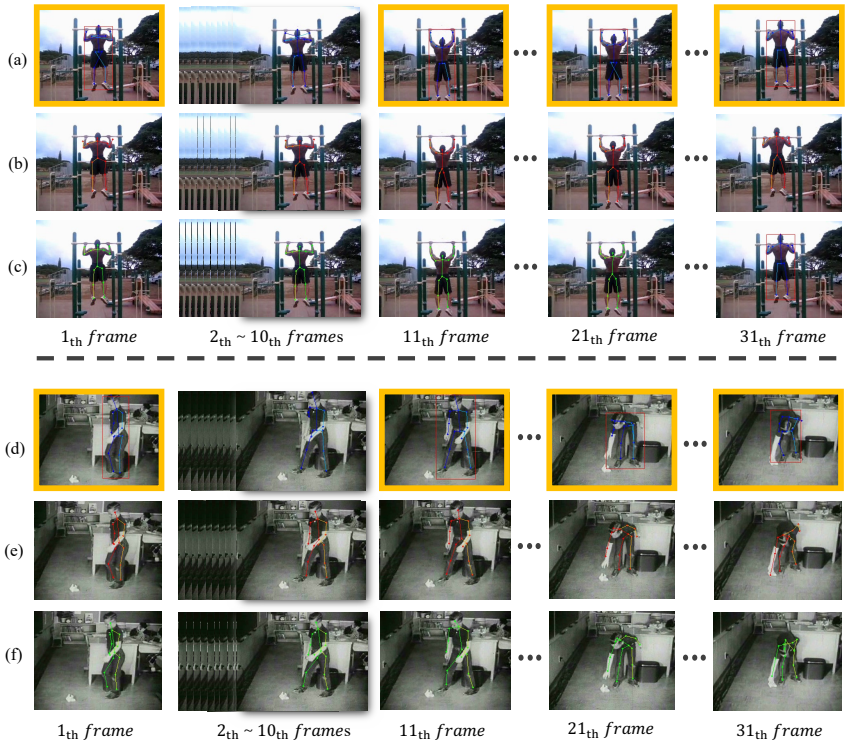


Fig. 3. Visualization the impact of denoising scheme in *DeciWatch* on calibrating the wrong detected poses on four visible frames from the single-frame backbone. We demonstrate the cases via two video sequences and simply ignore the invisible frames. (a) and (d) rows show estimated poses from the popular model SimplePose [14], where the sampling interval is 10. Inputs of *DeciWatch* are highlighted in the yellow boxes. (b) and (e) are output poses of our proposed *DeciWatch*, which can denoise and smooth the input poses by the proposed *DenoiseNet* and *RecoverNet*. (c) and (f) show the ground truth of the corresponding poses.

7 Failure Case Analyses

There are two types of failure cases in *DeciWatch*, which motivates the two corresponding future directions.

– *When the sampling rate is lower than the motion frequency of some body parts, it will be difficult to supplement the actual motion.* Human body is articulated. Thus different body parts have different movement frequencies and distribution. For example, the moving frequency and amplitude of hands and feet will be greater than that of the trunk. Our method adopts the same sampling rate for the whole body without considering that the motion distribution of different keypoints is different. In some actions, such as playing the guitar, only the hand will move at high frequency, but most other joints

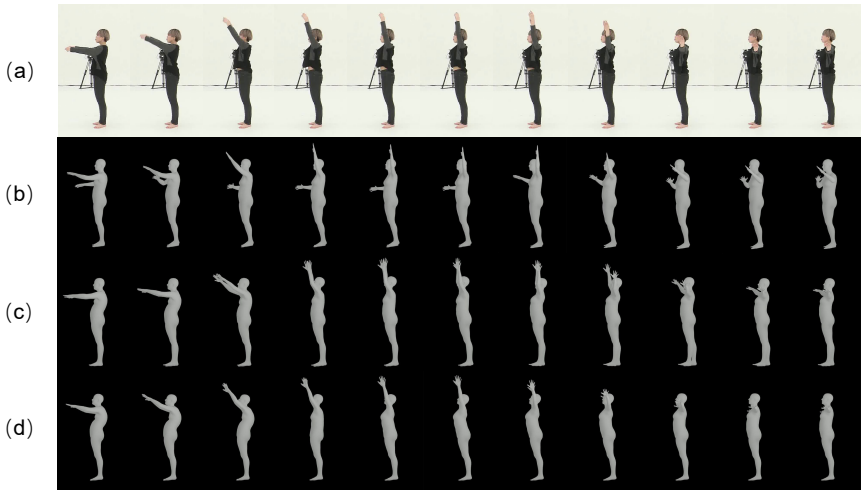


Fig. 4. Visualization of the recovery results on high-error estimated poses from AIST++ dataset. Only visible frames are shown, which are sampled with an interval of 10. Images in the row (a) are the original input frames at the 1_{th}, 11_{th}, 21_{th},...,101_{th} frame. (b), (c), (d) show the poses detected by SPIN [7], poses recovered by *DeciWatch*, and the corresponding ground truth.

will not move, so the detailed information recovery of hand movement will be lost. Therefore, adaptive sampling strategies, especially on different body parts or keypoints, will be beneficial.

– *If the estimated poses of most visible frames in the sliding window are in large errors, it is hard for* DeciWatch *to recover the correct poses.* As shown in Fig. 3, although our method can correct the noisy poses to some extent, this is the advantage of learnable methods. That is, the traditional interpolation method can not fix them. However, if most of the visible poses are noisy, our output may also tend to have similar (but smooth) errors. Thus, it is still essential to continuously improve the performance and robustness of pose estimation methods, especially in extreme scenes. At the same time, we can also consider using additional lightweight information, such as IMUs, to help improve performance.

References

1. Fan, Z., Liu, J., Wang, Y.: Motion adaptive pose estimation from compressed videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11719–11728 (2021)
2. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: 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)
3. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 3192–3199 (2013)
4. Joo, H., Neverova, N., Vedaldi, A.: Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In: 2021 International Conference on 3D Vision (3DV). pp. 42–52. IEEE (2021)
5. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7122–7131 (2018)
6. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: Pare: Part attention regressor for 3d human body estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11127–11137 (2021)
7. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2252–2261 (2019)
8. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13401–13412 (October 2021)
9. Luo, Y., Ren, J., Wang, Z., Sun, W., Pan, J., Liu, J., Pang, J., Lin, L.: Lstm pose machines. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5207–5215 (2018)
10. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 601–617 (2018)
11. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2640–2649 (2017)
12. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 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. pp. 7753–7762 (2019)
13. Tsuchida, S., Fukayama, S., Hamasaki, M., Goto, M.: Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In: ISMIR. pp. 501–510 (2019)
14. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European conference on computer vision (ECCV). pp. 466–481 (2018)
15. Zeng, A., Sun, X., Huang, F., Liu, M., Xu, Q., Lin, S.C.F.: Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In: ECCV (2020)

16. Zeng, A., Yang, L., Ju, X., Li, J., Wang, J., Xu, Q.: Smoothnet: A plug-and-play network for refining human poses in videos. arXiv preprint arXiv:2112.13715 (2021)
17. Zhang, Y., Wang, Y., Camps, O., Sznaier, M.: Key frame proposal network for efficient pose estimation in videos. In: European Conference on Computer Vision. pp. 609–625. Springer (2020)