3D Human Pose Estimation - A simple model for extending to video the regression of 3D pose from 2D pose

Victor SANH
victor.sanh@ens-paris-saclay.fr

January 21, 2018

Abstract

Martinez and al. introduced a surprisingly simple linear model to perform 3D human pose estimation from 2D pose estimates that compete with the best methods available in the literature. Building on the work of Martinez, I designed a simple model that try to extend the previous work to videos by taking into account the temporal continuity between consecutive frames. The model was trained and extensively tested on the Human3.6M dataset. Results show that this new model slightly improve the performance, but more importantly, it successfully capture the temporal continuity between frames as consecutive 3D predictions are significantly smoother. The manipulation video dataset was also used to perform quantitative comparison.

1. Introduction

3D human pose estimation is the problem of estimating the locations in the 3D space of specific joints that model our body. The vast majority of challenges in real life that involve 3D human pose estimation take as input images, videos or paintings and aim to give the 3D pose. A common approach to this problem is to decouple the estimation into two parts: from an image to the 2D joints, and from the 2D joints to the 3D joints. The 2D joints are basically the 3D joints that we have projected to remove one space component.

Martinez and al. [5] focused on the second part and proposed a simple linear neural network to infer the 3D joints from the 2D joints, the estimation of 2D joints from images being performed with another neural network (Stacked Hourglass) introduced by Newell and al. [6]. However, the proposed model of Martinez only takes single images to output single 3D poses. It does not take into account the temporal information and continuity that we have in videos. The goal of this project is to propose a model for including this temporal information.

First, I will present the model I designed based on the previous work of Martinez [5]. Then, I will show some quantitative results showing that the model presented perform slightly better than the basic one, and confirm this with visual examples. Last, I will present some quantitative results to compare the two models on some instances of the manipulation video dataset.

2. Bi-frame model

2.1. Uni-frame linear model by Martinez

The model presented by Martinez is a deep multilayer linear neural network with batch normalization, dropout, Rectified Linear Units (Relus) and residual connections that empirically improve the performance. The main block of the network is composed by the succession of two of these linear units, and this block is repeated twice with residual connection. Plus, there are two other linear layers: an input linear layer and an output linear layer that convert the input 2D pose and output 3D pose to the right size. The model is represented in Figure 1. The input of the network is of size $2N$ and the output is of size $3N$ where N is the number of joints that model our body. Note that a max-norm constraint of 1 was applied to the weights of the network.

2.2. An extension of the uni-frame model: bi-frame model

This simple model does not take into account the temporal relation between consecutive frames in a video. To capture the temporal continuity in a video, I changed the input size of the network to stack two consecutive frames. The input size is now of $4N$ which are the 2D joints of frames at time t and $t-1$. The output's size has not changed: the output is still the 3D joints at time t. Figure 2 shows a diagram of this approach.

2.3. Conventions and data processing

I kept the same conventions as Martinez: I used 2D and 3D joints rather than 2D probability distributions or 3D probability distributions. Their low dimensionality is an appealing feature to work with: for instance, one can easily

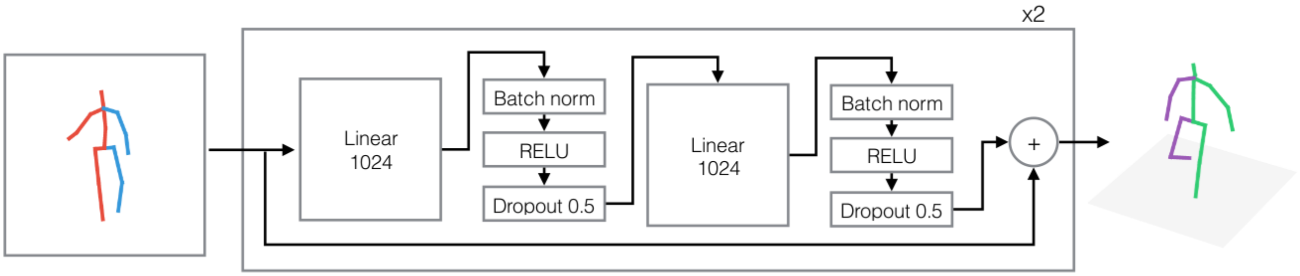

Figure 1. Structure of the model designed by Martinez. The input and output linear layers are not represented. The two main blocs are linked by a residual connection.

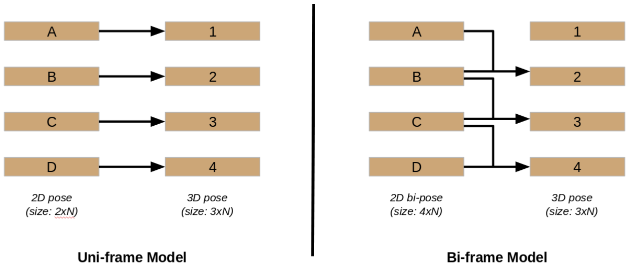

Figure 2. Schema of the bi-frame model versus the uni-frame model: it takes as input two consecutive frames

charge the whole preprocessed Human3.6M dataset[3] in memory while training the network. However, as the model I proposed suggest, we should increase the dimensionality of the input to get more accurate results.

Moreover, I applied normalization of the 2D and 3D joints: I subtracted the mean and divide by the standard deviation of each component. The network is thus trained with 2D and 3D normalized joints, and one should un-normalize the data after prediction by the network. In line with the Human3.6M protocol, I also zero-centered the 3D joints around the hip joint. The 3D joints are inferred in the natural coordinate frame of the camera, rather than a global arbitrary coordinate frame. This is performed by rotating and translating the 3D ground truth joints according to the inverse transform of the camera. Hence, it enables to consider all the cameras in the same way.

3. Training and testing uni-frame and bi-frame model

3.1. Dataset and protocol

I trained the bi-frame model on the Human3.6M dataset for which we have the 3D ground truth along with camera parameters and the sequences of images. I more specifically used the pre-processed data provided by Martinez along with the code of the model. Training was performed using subject 1, 5, 6 and 8, while testing was performed using subject 9 and 11. The error reported is the average error (in millimeters) across all 3D joints after alignment to the hip joint. In some baselines, the prediction has been further aligned by a rigid transformation (procrustes analysis) with the ground truth. I also used this rigid alignment in some of the tested models.

As suggested by Martinez, I trained the network for 200 epochs, using Adam method [4] to automatically optimize the learning rate with initial value of 0.001 and exponential decay. I set the batch size to 64 and initialize linear layers with Kaiming method [2]. I trained the network using a Geforce GTX 960M, a complete training takes 22 hours.

3.2. Quantitative comparison

I first tried to reproduce some of the numerical results reported by Martinez and al. I trained myself the model of Martinez and compared the test errors. Table 1 shows the details of these errors for each action. The lines in blue are the numbers reported by Martinez for which I obtained similar results (± 0.2mm). Note that for the model *Uni-frame - LB = 2; LU = 1024; 2D Joints = GT; RA*, I obtained slightly better average error 33.8mm (compared to 37.1mm reported in Table 1 of Martinez's paper [5]). To train the bi-frame model, I only used the ground truth 3D joints as they lead to much better performance.Moreover, Martinez reported a reasonable response when using a network trained on ground truth 3D joints and applied on noisy 2D detections.

Several lessons can be pointed out from these figures: the error is slightly improved with the bi-frame model (33.7mm compared to 31.4mm). The improvement is comprised between 1.5mm (*Purchases*) and 3.0mm (*WalkTogeter*). Most of the improvement seems to come from the model, and not the parameters I played with: I mainly played with the number of linear blocks, the procrustes analysis and the number of linear units per layer. Increasing the number of linear blocks from 2 to 3 only improves the average error of 0.2mm, increasing the number of linear units per layer from

Model	*Direct*	*Discuss*	*Eating*	*Greet*	*Phone*	*Photo*	*Pose*	*Purch*	*Sitting*	*SittingD*	*Smoke*	*Wait*	*WalkD*	*Walk*	*WalkT*	**Avg**
Uni-frame - LB = 2; LU = 1024; 2D Joints = SH; RA	42.2	48.0	49.8	50.8	61.7	60.7	44.2	43.6	64.3	76.5	55.8	49.1	53.6	40.8	46.4	52.5
Uni-frame - LB = 2; LU = 1024; 2D Joints = SH	53.3	60.8	62.9	62.7	86.4	82.4	57.8	58.7	81.9	99.8	69.1	63.9	67.1	50.9	54.8	67.5
Uni-frame - LB = 2; LU = 1024; 2D Joints = GT; RA	28.4	33.8	30.8	33.7	33.2	39.9	34.1	30.1	38.4	42.5	35.5	33.7	35.5	26.7	31.1	**33.8**
Uni-frame - LB = 2; LU = 1024; 2D Joints = GT	37,7	44,4	40,3	42,1	48,2	54,9	44,4	42,1	54,6	58.0	45,1	46,4	47,6	36,4	40,4	45,5
Bi-frame - LB = 2; LU = 1024; 2D Joints = GT; RA	28,8	33,6	30,4	33,8	33,1	39,8	33,3	30,1	38,2	42,1	34,3	33,7	35,7	26,2	30,2	33,5
Bi-frame - LB = 2; LU = 1024; 2D Joints = GT;	37,3	43,2	36,1	40,9	51,8	49,2	44,6	39,0	49,0	50,3	41,2	42,1	41,9	37,7	33,8	41,5
Bi-frame - LB = 2; LU = 1536; 2D Joints = GT; RA	27,0	31,6	28,0	31,9	30,7	37,5	32,0	28,6	35,7	38,7	31,9	31,5	32,6	24,6	28,0	**31,4**
Bi-frame - LB = 2; LU = 1536; 2D Joints = GT;	37,3	43,2	36,1	40,9	51,8	49,2	44,6	39,0	49,0	50,3	41,2	42,1	41,9	37,7	33,8	41,4
Bi-frame - LB = 3; LU = 1024; 2D Joints = GT;	35,1	41,9	37,5	40,0	42,3	50,9	42,4	38,3	49,3	51,1	40,8	40,6	42,3	31,6	34,9	41,3

Table 1. Detailed Results on H3.6M dataset. Several models are trained and tested. LB denotes the number of linear blocks, LU the number of linear units per layer, RA denotes if the predictions were aligned with the ground truth through procrustes analysis. The 2D joints used for training and testing were either the ground truth 2D joints (GT) or the 2D detections of Stacked Hourglass (SH). The lines in blue are the ones reported by Martinez.

1024 to 1536 only improves the error of 0.1mm. As expected, the procrustes analysis improves the performance, but it seems a little bit unfair in the sense that we do not always have the ground truth.

3.3. Qualitative comparison on single frames

We show some qualitative comparisons on single frames. As said previously, the improvement on the error is limited so the difference between the following images can be not obvious at first. Figure 3 shows an example of 3D outputs that are visually pretty much the same: most of the samples are similar between uni-frame and bi-frame model prediction.

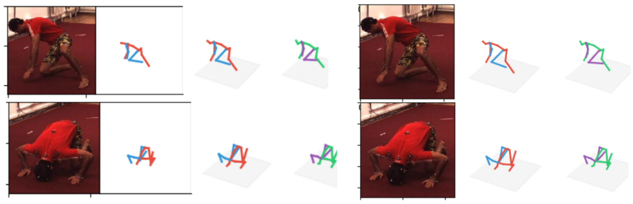

Figure 3. On the left: uni-frame model (from the left to the right: raw image, 2D detection using Stacked Hourglass, 2D ground truth, 3D estimate). On the right: bi-frame model (raw image, 3D ground truth, 3D estimate). Red and green indicate a left part while blue and purple indicate a right part.

Some improvement cases may be observed from single images estimate. Figure 4 displays an example of these cases. Empirically, these improvement cases seem to happen when the two consecutive frames are almost the same, that is to say when there is no much movement between two consecutive frames. Hence, the bi-frame network is fed with very similar information twice and seems to better estimate the 3D pose.

As suggested by Table 1, both models are really sensible to the quality of the 2D joints estimation. Figure 5 shows an example of these cases for both models in which the 2D estimation failed and thus the 3D estimate fails too. Surprisingly, on average, the response to self-occlusion is reasonable in both models. Figure 6 shows examples for the actions *SittingDown* and *Phone* for which the errors reported for all models (in Table 1) are among the largest ones.

3.4. Qualitative comparison on videos

We now compare the uni-frame and bi-frame models on videos of Human3.6M dataset. The 3D predictions are simply concatenated through time to produce a video along with the raw video of the scene and the 2D estimates. The fps used is 50. The 3D predictions for the bi-frame model are significantly smoother than for the uni-frame model: the movements induced look less jerky and abrupt. Multiple videos are available at `https://goo.gl/G7Smj3` for subjects 9 and 11 and for different actions. Figure 7 shows the impact of the smoothness of the bi-frame model: when the 2D estimate fails, the impact on the bi-frame 3D estimate is delayed.

In some cases of complex position with high degree of self-occlusion, the output of the bi-frame loses its smoothness property mainly because of the poor quality of the 2D joint detection. One example is available at `https://goo.gl/G33SQS` between $t = 20s$ and $t = 22s$. Though, we notice that the gap of smoothness between uni-frame model and bi-frame model seems more significant on actions for which we reported a comparatively low error (such as *Direction*) and more subtle for actions that are more difficult (such as *Photo*).

4. Quantitative analysis on manipulation video dataset

I also performed quantitative study on the dataset manipulation video [1]. For this dataset, only 3 images per sequence are annotated with ground truth 3D positions. As commented previously, both uni-frame and bi-frame models are sensitive to the quality of the 2D estimations. I used the Stacked Hourglass network to predict 2D joints from images (the implementation I used is available here: `https://goo.gl/T15wKM`). This network is really sensitive to the input image, and the localization of the person in the image: it fails to detect 2D joints if the person is not at the center of the image and does not take the majority of the space. I cropped and centered all the images around the body before running Stacked Hourglass network so that the images

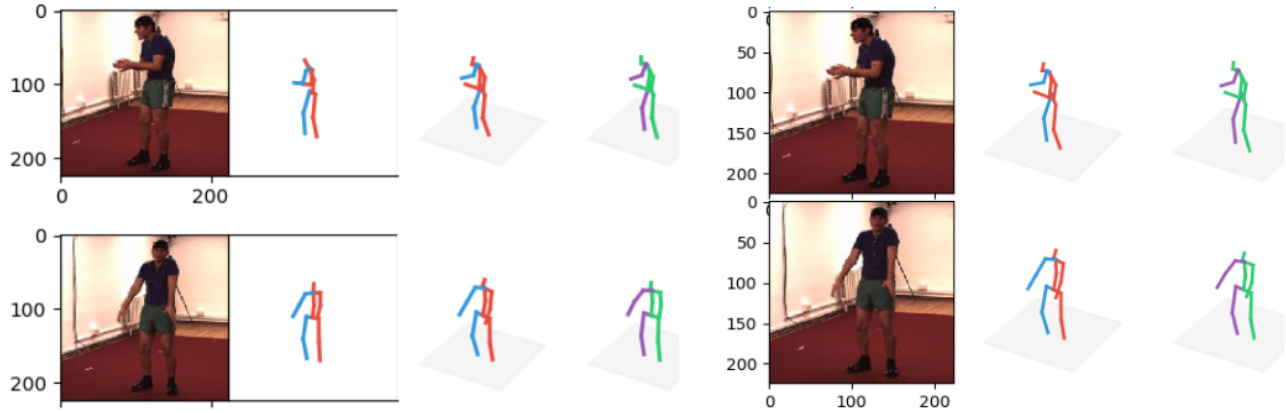

Figure 4. The 3D estimates of the bi-frame model (right) are slightly better on the left arm compared to the ground truth than the uni-frame model (left).

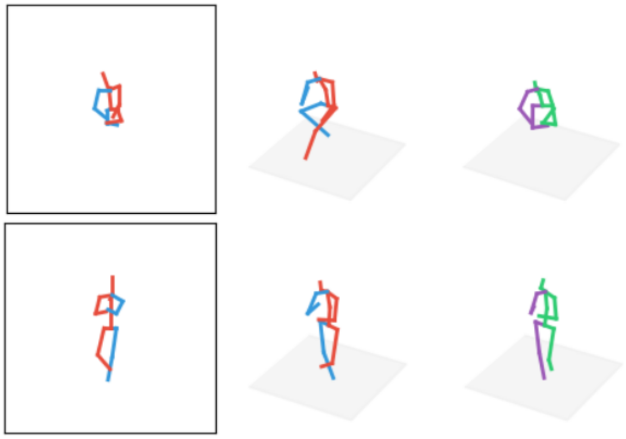

Figure 5. Sensibility to the 2D estimate for both uni-frame (top) and bi-frame (bottom) model. From the left to the right: 2D estimate using Stacked Hourglass, 3D ground truth, 3D estimate

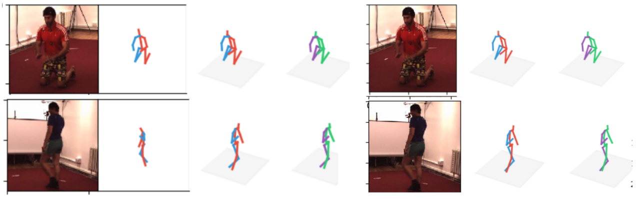

Figure 6. Reasonable response to self-occlusion

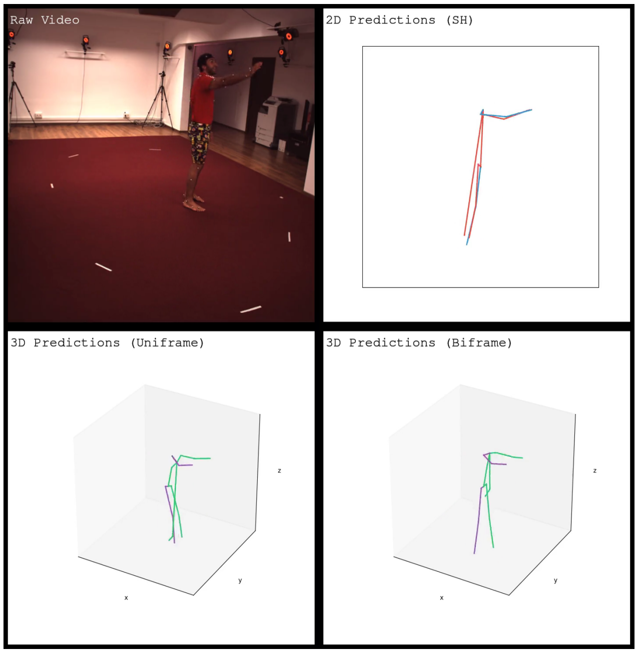

Figure 7. The 2D estimate fails and it impacts the uni-frame 3D estimate right away, whereas the bi-frame model 3D estimate seems more reasonable at this time t

given to the pipeline are easy to deal with (this might explain the surprisingly low error for some sequences, in fact the unit of measure may not be the millimeters...). Figure 8 shows an example of the prepossessing where Figure 9 is the initial image on which Stacked Hourglass failed to detect 2D joints. Table 2 shows the average errors across the three annotated images obtained for the uni-frame model. 3D predictions wetr aligned with the ground truth through procrustes analysis. The dispersion of errors is really high. This might be because of the high level of self-occlusion that sequences like Jackhandle 2 or Wheel 2 perform poorly (see Figure 11).

I also tried to use the bi-frame model on the same sequences, but unfortunately, due to an still unresolved bug, I was not able to output reasonable 3D predictions. I am still investigating the bug. As previously, I expect the errors to be slightly better when using two consecutive frames.

Sequence Name	Average Error (mm)
Barbell 1	11.8
Barbell 2	18.2
Hammer 1	17.2
Hammer 2	24.3
Jackhandle 2	44
Wheel 1	23.9
Wheel 2	25.2
Wrench 1	18.2
Wrench 2	25.5

Table 2. Errors for the uni-frame model on manipulation video dataset for several sequences.

5. Critics and future work

The network proposed by Martinez works with normalize data: we should normalize the data before feeding to the network and un-normalize the output of the network to get the 3D joints. I used the 2D and 3D mean and standard deviation computed on Human3.6M dataset. It is not a problem when working on Human3.6M data, but there is no reason that theses normalization parameters are still relevant for manipulation videos.

Section 3.4 suggests that the movement is a discriminative feature that helps to improve the error. I believe that replacing one of the consecutive frames in the bi-frame model by the optical flow beween frame t and $t{+}1$ computed in the locations of the 2D joints is an interesting lead. The input size would still be $4N$: $2N$ for the 2D joints of frame t and $2N$ for he optical flow computed between t and $t + 1$ and taken in the localization of the 2D joints of frame t. Hence, we can keep the same network architecture.

The smoothness observation deduced in section 3.4 also suggests that we can add more temporal context to improve the smoothness (rather than considering two consecutive frames, the low dimentionality of the frame representation enables us to take 3 or 4 consecutive frames).

6. Conclusion

I have built on the work of Martinez and al. to propose a simple bi-frame model that tries to take into account the temporal continuity in videos. Results show that not only this model slightly improves the 3D prediction error, but more importantly, it significantly enhance the smoothness of the 3D estimation between two consecutive frames. The code I developed during this project (mostly everything around the bi-frame model, numerical evaluationsa and some of the plots) is available on this repository: `https://goo.gl/3ckCwK`. This repository is based on the code furnished along with the paper of Martinez: `https://github.com/una-dinosauria/3d-pose-baseline`. Extraction of ground truth for manipulation video was performed using a code developped by Li Zongmian.

References

[1] Manipulation videos dataset. `https://www.dropbox.com/s/5608fx0p23jdvjp/manipulation_videos.zip?dl=0`.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[3] C. Ionescu, D. Papava, V. Olaru, and C. 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.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. *CoRR*, abs/1705.03098, 2017.

[6] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.

Appendix

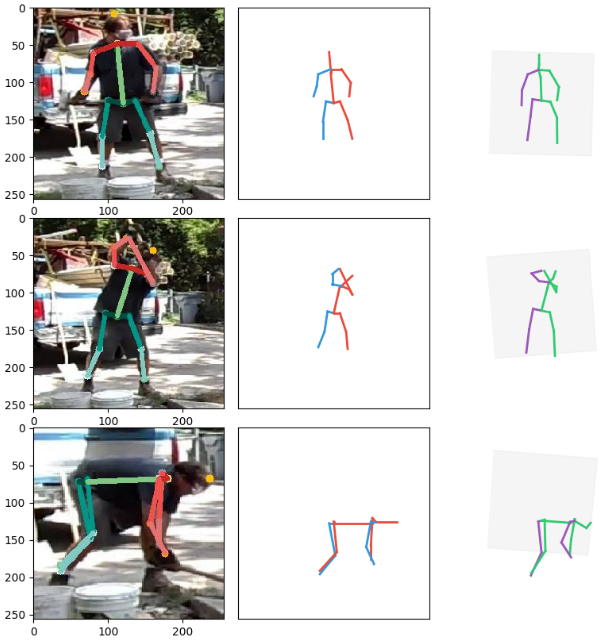

Figure 8. Sequence *Hammer_0001* from Manipulation video dataset. From the left to the right: preprocessed image with SH detection, 2D estimate, 3D estimate. Error from the top to the bottom (in mm): 11.6, 10.2, 29.7

Figure 9. Initial image (before centering and cropping) of the top detection in Figure 8

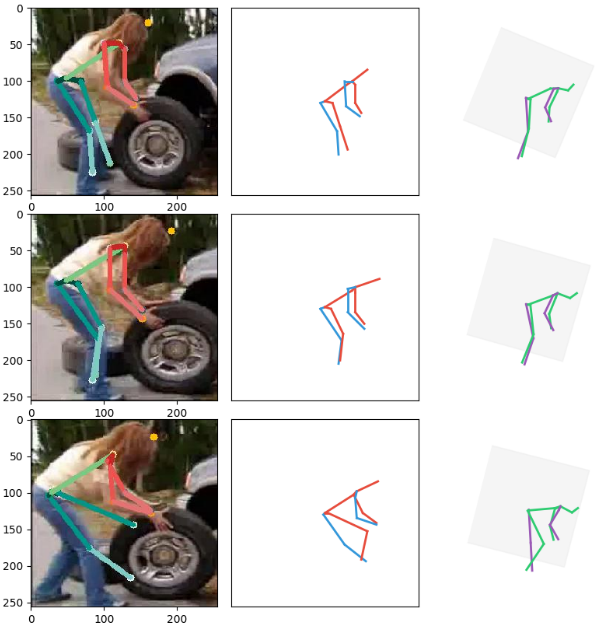

Figure 10. Sequence *Wheel_0002*. The SH detection failed even after preprocessing. Orientation of the 3D estimate was changed for better point of view as might not exactly correspond to the one of 2D estimate. Error from the top to the bottom: 20.7, 21.2, 33.7

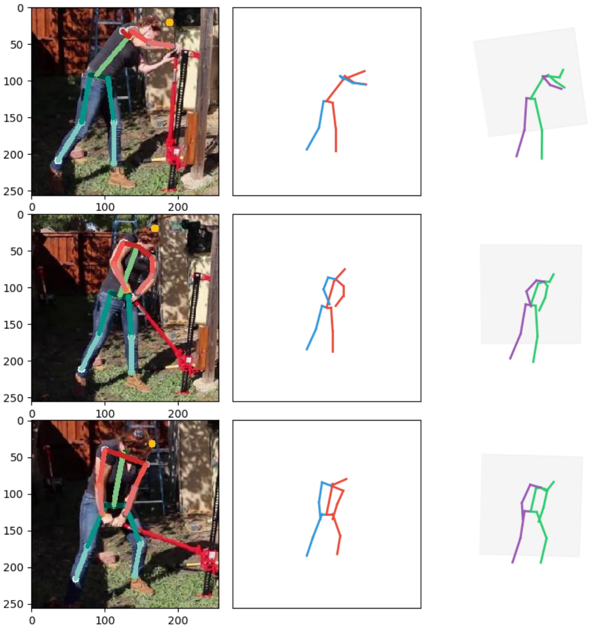

Figure 11. Sequence *Wrench_0001*. Error from the top to the bottom: 19.8, 17.4, 17.4