ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
SCHOOL OF LIFE SCIENCES

Master project in Life Sciences Engineering

# 3D Pose Based Motion Correction for Physical Exercises

Done by
**Hugues Vinzant**

Under the direction of Prof. Pascal Fua
In the Computer Vision Laboratory (CVLAB)
EPFL

External expert
Lengagne Richard

LAUSANNE, EPFL 2021

# Contents

# 1 Summary

**With the rise of self-management for treatment of musculoskeletal disorders and especially during these times of pandemic, people tend to exercise alone and without supervision. Recent progresses in fields of pose estimation, action recognition and motion prediction allow us to analyze movements in details and thus identify potential mistakes done while exercising. In this work, we prepare a dataset containing videos, 2D and 3D poses of correct and incorrect executions of different movements that are SQUATS, lunges, planks and pick-ups and labels identifying the mistake in each practice of that exercise. This dataset is used to demonstrate our motion correction model, designed using a graph convolutional network architecture and trained with a differentiable dynamic time warping loss. As a result we are able to correct movement mistakes in 3D pose sequences and output the corrected motion. This model is integrated in a pipeline containing a state-of-the-art 3D human pose estimator to go from raw video images to a sequence of corrected 3D poses. Evaluation of this model is done using an action recognition model trained on the same dataset to recognize whether the sequence is correct or has a particular type of mistake. Results show that our model is successful in correcting incorrect sequences, as most of the time the resulting motions are classified as correct.**

# 2 Introduction

Worldwide, musculoskeletal disorders (MSDs) are a major burden affecting negatively everyday life and also are the main cause of disability among adults [34]. No matter if it is a lower back pain, neck pain or any other MSDs, the treatment often fails to efficiently reduce the pain and to get the patient adhesion. This is because most of current solutions aim for fast improvements in a very generic way and without considering patients implication or personal shortcomings for specific types of exercises [24]. Durable results can only be achieved through a patient-centered approach in the long-term focusing on self-management and healthier lifestyle [16].

Self-management can be defined as "The ability to manage the symptoms, treatment, physical and psychosocial consequences, and lifestyle changes inherent in living with a chronic condition." [2], meaning that patients must have an active role in their own way toward healing and that results are not expected to be immediate. Of course, patients do not have to follow this path alone and physical therapists are there to help them design and support their self-management strategies. On top of identifying the cause of the condition, giving advises on activities to perform differently or to avoid, the therapist can also prescribe therapeutic exercises improving performance and mobility [19]. Such exercises are also often performed by athletes during training and can even be useful for rehabilitation after injuries. However, bad realization of these exercises might cause more harm than good and sometimes even lead to injuries [20, 11]. Thus, it is crucial for the patients performing physical exercises at home and/or without the supervision of a specialist to get feedback regarding the correctness of the movements performed. This aspect is even more highlighted nowadays with the currently ongoing COVID-19 pandemic. Because of lockdowns and home-office policies, it might be hard to reach a physiotherapist or a gym and people tend to exercises less and less. Moreover, when they do, it is without supervision.

In this frame, the present project aims to develop a motion correction algorithm that, given a movement, is able to output a corrected version of it. This enables the patient to realize their mistake by looking at the correction and see how they can improve their practice. To do so, we prepare a dataset containing correct and incorrect versions of the same movement. We then use this dataset to develop a motion correction model and evaluate it thanks to an action recognition model trained using the same dataset. Finally, we integrate our motion correction model in a pipeline going from raw video images to corrected movement. To our awareness, correcting motion sequences in such a way has not yet been proposed.

# 3 Related Works

## 3.1 Action Recognition and Motion Prediction

Our motion correction task lies at the the interface of computer sciences and human motion analysis. This is also the case of the two other tasks that are action recognition (AR) and motion prediction (MP). As in our case, they take as input a sequence poses (or images) and aim to analyze it, whether to assign it a label in the case of AR or to forecast future body poses in the case of MP.

Most of the time, AR is done to discriminate between a limited number of actions that are often easily distinguishable one from another for a human eye. For example, UCF101 [37] and HMDB-51 [23] datasets actions can be sports (horse riding, skiing, rowing, ...), everyday activities (brush hair, type, cooking, ...), simple movements (clap, throw, turn, ...) and many others. In our case however, we want to discriminate between different realizations of the same action. Since this is a subset of actual AR task, it is still worthwhile to gain insight from recent state-of-the-art publications in that field.
In their paper from 2020, Esat Kalfaoglu et al. [8] use a 3D Convolutional Neural Network (CNN) architecture, but unlike previous paper implementing such architecture [5, 14, 39, 9] they replace the Temporal Global Average Pooling (TGAP) layer after 3D convolutions by a Bidirectional Encoder Representations from Transformers (BERT), thus achieving top-1 accuracy in both aforementioned datasets.
The same year, Gowda et al. [12] implemented a new algorithm for frame selection. It has been shown that better classification results can be achieved by making an optimal selection of frames rather than using the entire video [15, 36]. Gowda et al. implementation not only considers the importance of a frame but takes also into account its relation with other frames of the video thank to an attention model. When used as preprocessing, this model has been shown to improve state-of-the-art accuracy on UCF101 and HMDB datasets.
Another way to improve AR performance has been recently investigated by Li et al. [25]. In this work they show that fusing multiple modalities, and in particular pose information, can improve classification accuracy and even reach state-of-the-art results.

AR does not necessarily work with pose information, MP on the other hand needs a structured input in order to output something of the same shape. In that sense, this task is closer to ours. Classic architectures for MP are often based on Recurrent Neural Networks (RNNs) [10, 18, 32], but they suffer from discontinuities and mean pose convergence in the long-term. To counter that, other architectures are also explored and give equivalent or better results such as Generative Adversarial Networks (GANs) [13] or reinforcement learning [42]. In their two papers, Mao et al. [31, 30] propose a Graph Convolution Networks (GCN) structure encoding the spatial representation of the motion. [30] improves upon the state-of-the-art scores of [31] by adding an attention mechanism capturing motion redundancies. Due to their successful handling of 3D human motion sequences, we have based our motion correction architecture as a GCN as proposed in these works.

## 3.2 Exercise Correction

Numerical automatic exercise correction task have been benefiting from recent progress in pose estimation to move from the use of multiple sensors such as accelerometers or gyroscopes to the use of cameras alone. Some applications aim to recognize fitness exercises and count the number of repetitions [21, 1], however they have not yet attempted to provide feedback and correction on the exercise execution. Differences between a target pose and the one of the subject doing the exercises have been studied both for 2D [38] and 3D [43] poses. However, these techniques only work for static poses such as yoga or Tai-chi poses but often fails for active motions as we often encounter in fitness. To overcome this, Chen et al. [6] and Liu et al. [26] developed pipelines able to evaluate whether a movement is correctly executed or not. They both also indicate why the posture incorrect-

ness happens. However while the last two papers are offering understandable feedback, it is only based on few fitness indicators such as angle or distance between body segments meaning that they are not able to detect or provide feedback for incorrect movements that are not corresponding to any registered indicator. Moreover, both methods are working with 2D poses which lacks depth information that can be crucial in human pose. We thus propose a correction model working with 3D poses and able to correct a movement without predefined heuristic rules on angles or distances.

# 4 Methods

## 4.1 Dataset

In the frame of this project, we have formed a new dataset containing both correct and incorrect versions of a physical exercise, both executed by the same subject. The following sections explain how such data is acquired, refined and labelled.

### 4.1.1 Data acquisition

As shown in figure 1, the data is acquired in a room equipped with 4 GOPROs approximately positioned on a circle around the subject performing the actions standing in the middle. The cameras are oriented so that the subject appears in the middle of the image. The sampling rate is 30 frames/second and the images have a size of 1920x1080 pixels.

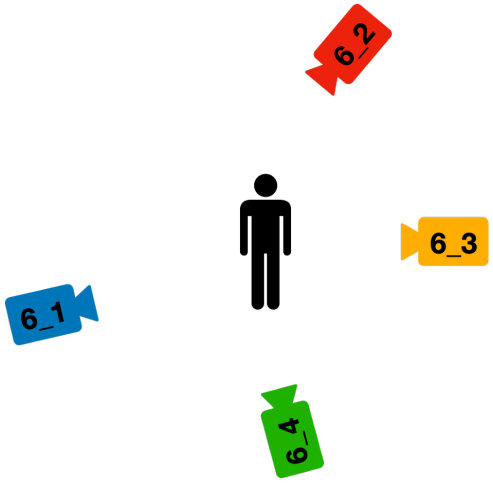


**Figure 1: Acquisition setup.** Cameras are placed around and oriented toward the subject performing the action.

The list of actions to be performed by the subject, as well as the instructions and the number of repetitions, are summarized in table 1. This process is repeated for 4 different subjects but all the data is acquired the same day under the same conditions. SQUATs, lunges and planks are strength exercises. A SQUAT consists in lowering the hips from a standing position and standing back up. A lunge refer to a position in which one leg is moved forward with bent knee and foot flat on the ground while the other leg stays behind. The plank is the action of maintaining a straight position with hands and toes on the ground. Sketches of these three actions are visible in figure 2. Pick-up refers to the action of bending to pick a small cube lying on the ground, standing up with it and putting it back down.

| Action | Instruction | Repetitions |
|---|---|---|
| SQUATs | Correct | 10 |
| | Feet too wide | 5 |
| | Knees inward | 5 |
| | Not low enough | 5 |
| | Front bent | 5 |
| Lunges | Correct | 10 |
| | Not low enough | 10 |
| | Knee passes toe | 10 |
| Planks | Correct | 10 |
| | Arched back | 10 |
| | Hunch back | 10 |
| Pick-ups | Correct | 10 |
| | Straight legs | 10 |
| | Asymmetric | 10 |

**Table 1: Action instructions.** List of actions as well as the instructions and the number of repetitions given to each participant.

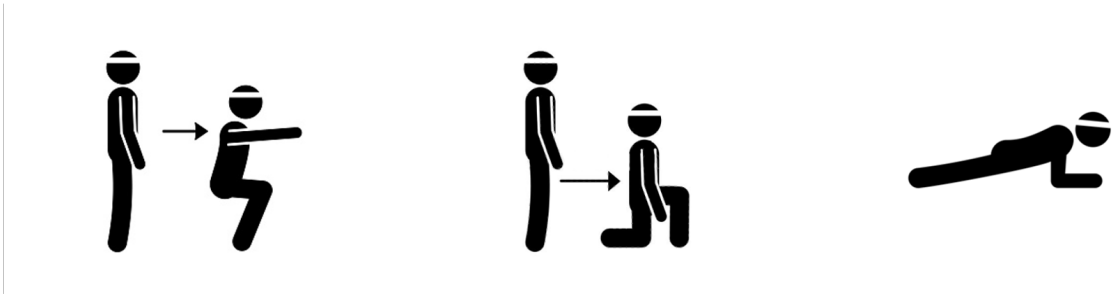

**Figure 2: Actions.** Three first actions asked to be performed by the subjects. From left to right SQUAT, lunge and plank.

Once the data is acquired for every subject, videos from cameras 6_1, 6_2, 6_3 and 6_4 are synchronized. In order to have a clear signal when to cut the videos so they start at the exact same frame, the first subject was asked to clap in his hands before following the experiment instructions (similar to the use of the clapperboard in the film industry to synchronize image and sound). The clapping frame of each camera is selected as the starting point of the recordings. Then, we manually label groups of frames corresponding to the same subject, action and instruction and extract them using the ffmpeg project.

### 4.1.2 Camera parameters

In order to compute the ground-truth human 3D poses of the sequences we first find the camera parameters. Camera intrinsics (optical center and focal length) and distortion coefficients of the specific cameras used were already available from previous experiments conducted in the CVLAB. Camera extrinsics, which indicate the location and rotation of the camera in the 3D scene, are relative to the experiment and thus needed to be computed. In our case, since the cameras were static during all the recordings, this is done only once according to the following steps. First, we find and annotate static landmarks in the first frame of each camera, such as cables or tape marks on the floor. Then, we extract the 2D poses of five randomly selected frames using the keypoints detector OpenPose [4] and use these keypoints as mobile landmarks. This outputs 25 keypoints

(or joints) described in figure 3. In some cases, a joint can be hidden from a specific view point (e.g. The left elbow can be hidden behind the subject chest if the camera is looking from the right side) and in such conditions, the software might be uncertain about the exact location of the joint. Luckily, OpenPose also outputs a confidence score for each keypoint prediction, allowing us to only select the keypoints having a confidence score above 0.9. Once all static and mobile landmarks have been identified, these are combined for each frame and fed to the camera calibration code obtained from the CVLAB using landmarks to compute 3D locations and rotations of the cameras in a specific frame. The code also refines these values using bundle adjustment which minimize the sum of squares reproduction error between predicted points and original position in the image by using least squares [40]. Finally, camera extrinsics for the entire experiment are obtained by averaging the ones of the five frames for which mobile landmarks were computed.

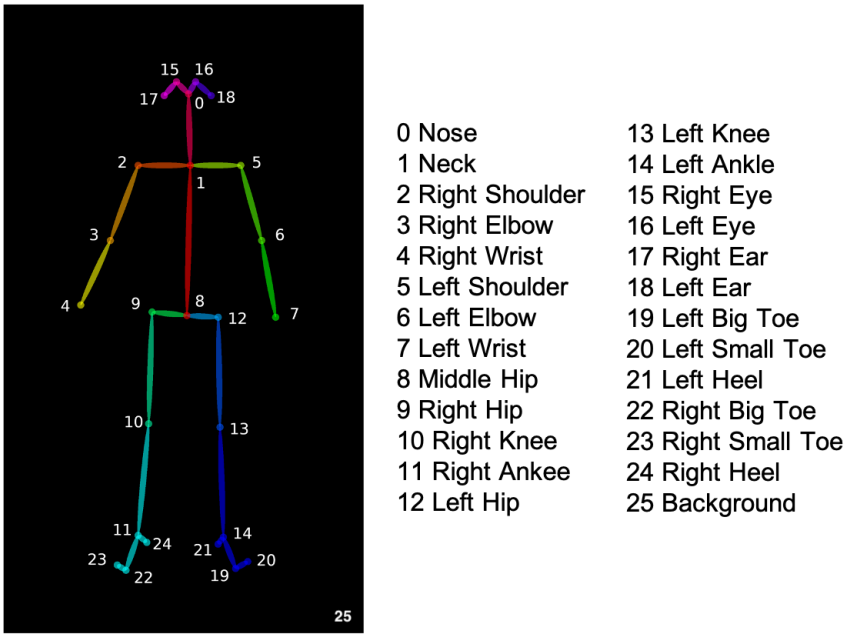


**Figure 3: OpenPose 25 keypoints.** Pose output format and joint ordering from [4].

### 4.1.3 Pose estimation

To obtain 3D ground truth poses of the subject at every frame, we start with computing 2D poses from the four views using OpenPose [4] as in section 4.1.2. Even if the cameras are not particularly wide angle, we undistort the points using cameras intrinsics. Then, we remove outlier poses by computing the euclidean distance between the location of a joint and the location of the same joint in the previous and the following frame. If this distance is above a certain threshold, the location is set as being the mean of the previous and the following ones. Then, for each pair of camera we compute the 3D positions of the 2D points appearing in both views using point triangulation. If one camera gives bad estimations we remove view pairs containing this camera and set the 3D position as the mean over all remaining pairs. Outlier poses are also corrected in the same way as for 2D poses and sequences of poses are temporally smoothed by convolution with a hamming filter. Size 11 is chosen as it represents approximately $1/3^{rd}$ of a second at 30fps. Finally, in order to obtain the 2D poses corresponding to our 3D ground truth, 3D poses are projected back on each camera. The dataset obtained is finally stored in a dictionary structure illustrated in listing 1.

```
params:
    camera:
        intrinsics: {K, d}
        extrinsics: {R, t}
frames:
    action:
        subject:
            instruction:
                frame #:
                    path:
                        camera: '/path/frame.png'
                    2D openpose:
                        camera: joints
                    2D projection:
                        camera: joints
                    3D ground truth: joints
```

**Listing 1:** Python dictionary structure used to save the data in a pickle file.

#### 4.1.4 Individual movements

Until this point, frames are grouped depending on the action, the subject and the instructions in table 1. Here, we proceed to isolate individual repetitions. For SQUATs and lunges, this is done by taking the distance between the farthest joints vertically (x-coordinate) on the 2D pose. Assuming that these actions start and end in standing position, these two joints are most likely located on the head and on the feet. Regarding the plank exercise, we take the knees height first discrete derivative along time as separation criterion since subjects are putting their knees down in between each action. These values are plotted against time and mean filtered. Then by cutting the signal at each peak it is possible to separate individual actions. Filter size and peak finder parameters are summarized in table 2. On the other hand, for the pick-up exercise it is not possible to separate individual movements because subjects are actually performing the task in different ways (i.e. some are keeping the cube in their hands the whole time while others put it down in between each movement). Therefore, we decide not to label these and to continue only with SQUATs, lunges and planks.

| Action | Filter size | Minimum peak height | Minimum peak distance |
|---|---|---|---|
| SQUATs | 20 | 550 | 45 |
| Lunges | 20 | 550 | 60 |
| Planks | 15 | 1.5 | 100 |

**Table 2: Peak finding parameters.**

Once all repetitions are isolated, all 3D poses are mapped to the same default skeleton (being the one of the first subject). Poses are thus normalized in order for the middle hip to be at the center of the scene. Then, starting from this joint (and until all joints were reached), linked joints are moved closer or farther so the link length is the same as the one of the default skeleton but keeping the same angles. Finally, the skeleton is oriented so that the shoulder vector (between left and right shoulders) lies on the x-axis and the spine vector (between neck and middle hip) lies on the y-axis.

## 4.2 Motion correction

### 4.2.1 Ground truth

Before training any motion correction architecture, the dataset obtained and cleaned in previous sections is reshaped to an array of size 3J x L where J is the number of joints selected and L the number of frames in the sequence and 3 represents the xyz coordinates. At this point, some joints (eyes, ears and small toes) are removed as we expect them to have no or very little influence on the motion correctness. Also, each incorrect action is associated with a correct one as ground truth. Since sequences do not necessarily have the same length, they are first aligned using 2D Dynamic Time Warping (DTW). This algorithm has been introduced in 1959 [3] and has been used in very different fields since then, including computer vision and motion analysis [35] [44]. Given two sequences $S_x \in \mathbb{R}^{3J \times L_x}$ and $S_y \in \mathbb{R}^{3J \times L_y}$, DTW finds an optimal match constrained by the fact that every index from one sequence must be matched with one or more from the other, first and last indices from both sequences must be matched together (but this does not have to be their only match) and the mapping has to be monotonically increasing meaning that if the $i^{th}$ point of one sequence is mapped to the $j^{th}$ point of the second sequence, points $> i$ can only be mapped with points $\geq j$. Pairing is then done by assigning to each incorrect action, the correct action corresponding to the same activity and to same subject and having the lowest Euclidean distance computed on DTW-aligned sequences.

### 4.2.2 Model

As seen in the previous section, motion sequences can have different lengths. In order to counter that, we modify the inputs when feeding our data to our model by using 2D Discrete Cosine Transform (DCT) following the implementation described in [29]. Taking K = 25, the new sequence $\mathbf{X} = [x_0, ..., y_K] \in \mathbb{R}^{3J \times K}$ of a signal $\mathbf{S} = [s_0, ..., s_L] \in \mathbb{R}^{3J \times L}$ is computed as

$$x_k = \sum_{l=0}^{L-1} s_l \cos\left[\frac{\pi}{L}\left(l + \frac{1}{2}\right)k\right], \tag{1}$$

and the original signal can be recovered by

$$s_l = \frac{1}{\sqrt{2}}x_0 + \sum_{k=1}^{K-1} x_k \cos\left[\frac{\pi}{L}\left(l + \frac{1}{2}\right)k\right], \tag{2}$$

Note that equations 1 and 2 are respectively the DCT-II and a scaled DCT-III. Also, if $L \leq K$, the representation is lossless. On the other hand, if $L > K$, the mapping removes jittery motion frequencies.

The model architecture is widely inspired from [31] and as illustrated in figure 4 is composed of one Graph Convolutional Layer (GCL) for both input and output as well as two residual blocks, each composed of two GCLs. For training, and except for the last layer, all GCLs are followed by Batch Normalization (BN) and Dropout layers. Hidden size of 128 and ReLU activation function are chosen.

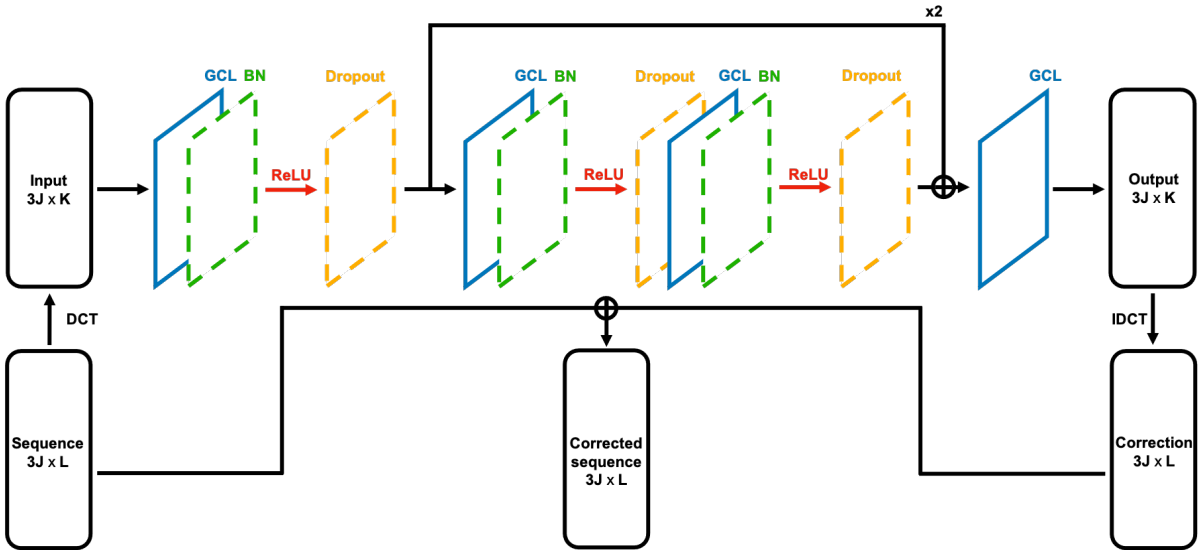


**Figure 4: Correction model architecture.**

GCLs are taking advantage of data structure by linking the nodes (here joint coordinate) thanks to a weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{3J \times 3J}$ resulting in the following output:

$$\mathbf{S^{(i+1)}} = \sigma\left(\mathbf{A^{(i)}}\mathbf{S^{(i)}}\mathbf{W^{(i)}}\right), \tag{3}$$

where $\mathbf{A^{(i)}}$, $\mathbf{S^{(i)}}$ and $\mathbf{W^{(i)}}$ are respectively the adjacency matrix, input and trainable weights of layer $i$ and $\sigma(x)$ is the activation function. In our case and as in [31], we do not set a predefined graph connectivity but we learn the adjacency matrix during training as we do for the weights. This model is trained for 150 epoch with a batch size of 128 and a learning rate at epoch i defined by the following equation:

$$lr_i = 0.01 * 0.9^{i//5} \tag{4}$$

Subjects 0, 1 and 2 were included in the training whereas subject 3 was kept out for testing.

As suggested by the architecture, the output has the same shape as the input and represents the correction to be made for each joint coordinate. this output is first mapped back to a signal of the same shape as the original sequence thanks to equation 2 and both are summed. Then to compute the loss between this corrected sequence and the ground truth it is paired with, we use a so called soft-DTW loss. This loss was proposed and implemented in [7]. It is built upon classical DTW but is made differentiable by computing the soft-minimum of all alignment costs.

### 4.2.3 Pipeline

Since 3D poses are not always available and that we want to correct movement based on videos, we integrated our work presented in previous sections in a pipeline going from raw video input to a corrected sequence of 3D poses. In order to do so, we use the official implementation of a state-of-the-art 3D pose estimator called "Video Inference for Body Pose and Shape Estimation" (VIBE) [22]. Their architecture takes a sequence of frame as input, feeds it to a Convolutional Neural Network (ResNet50), two Gated Recurrent Units layers and a linear projection layer. To get the pose and shape parameters, the obtained output is then fed to a regressor outputing the SMPL (Skinned Multi-Person

Linear Model) parameters [27]. The SMPL body model is then converted in the same shape as the 3D skeleton data displayed in figure 3 and the sequence is then fed to our motion correction model. Because VIBE is a multi-person pose estimator and that other persons than the subject sometimes appear in the videos of the dataset, we create rules to select the right subject. We select the subject as being the leftmost person for camera 6_4 and the rightmost person for camera 6_2. For camera 6_1 and 6_3, the subject is taken as the tallest person (i.e. the one with the largest bounding box).

### 4.3 Action recognition

In order to have an idea of the performance of our motion correction model, we propose here to create a motion classifier able to discriminate between action type and movement correctness. The architecture of such model is presented in figure 5 and is clearly inspired from the one for the correction without the residual blocks, a linear output layer with a LogSoftMax activation function and hidden size of 32. Again, this architecture was trained on subjects 0, 1 and 2 for 50 epoch with a batch size of 128 and a learning rate still defined by equation 4.

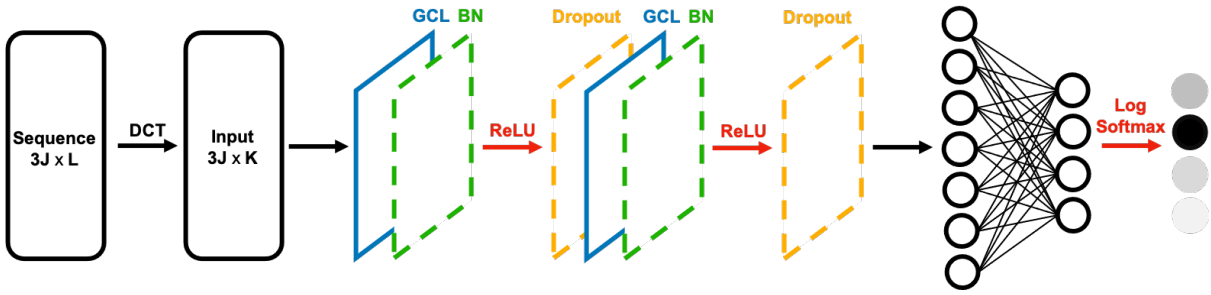


**Figure 5: Classification model architecture.**

## 5 Results

### 5.1 Dataset

After the experiments, each camera footage resulted in three 14 min videos, which were then concatenated to obtain in total four videos of approximately 45 min, organized as depicted in figure 6. Examples of acquired images for each action, subject and camera can be found in figure 7.

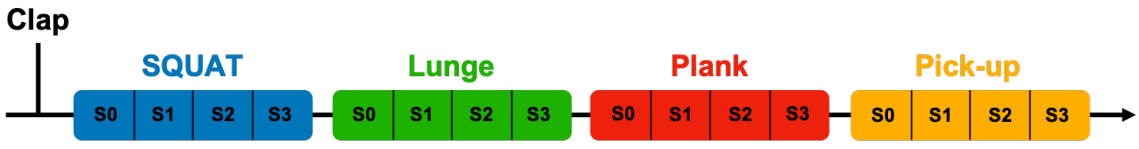


**Figure 6: Video content timeline.** The clap is used to synchronize all 4 videos. S0, S1, S2 and S3 respectively represent subject 0, 1, 2 and 3 performing all the instructions from the specified action as in table 1


**Figure 7: Samples images.** Images of 4 subjects, 4 actions and 4 cameras: **(A)** Subject 0 performing a SQUAT viewed from camera 6_1, **(B)** Subject 1 performing a lunge viewed from camera 6_2, **(C)** Subject 2 doing the plank viewed from camera 6_3, **(D)** Subject 4 picking-up the box viewed from camera 6_4.

Examples of static and mobile landmarks used to compute camera extrinsics for the first frame after synchronisation of the 4 cameras are displayed in figure 8. As one can notice, static landmark are mostly lying on the ground whereas mobile ones correspond to the most confident subject joints.

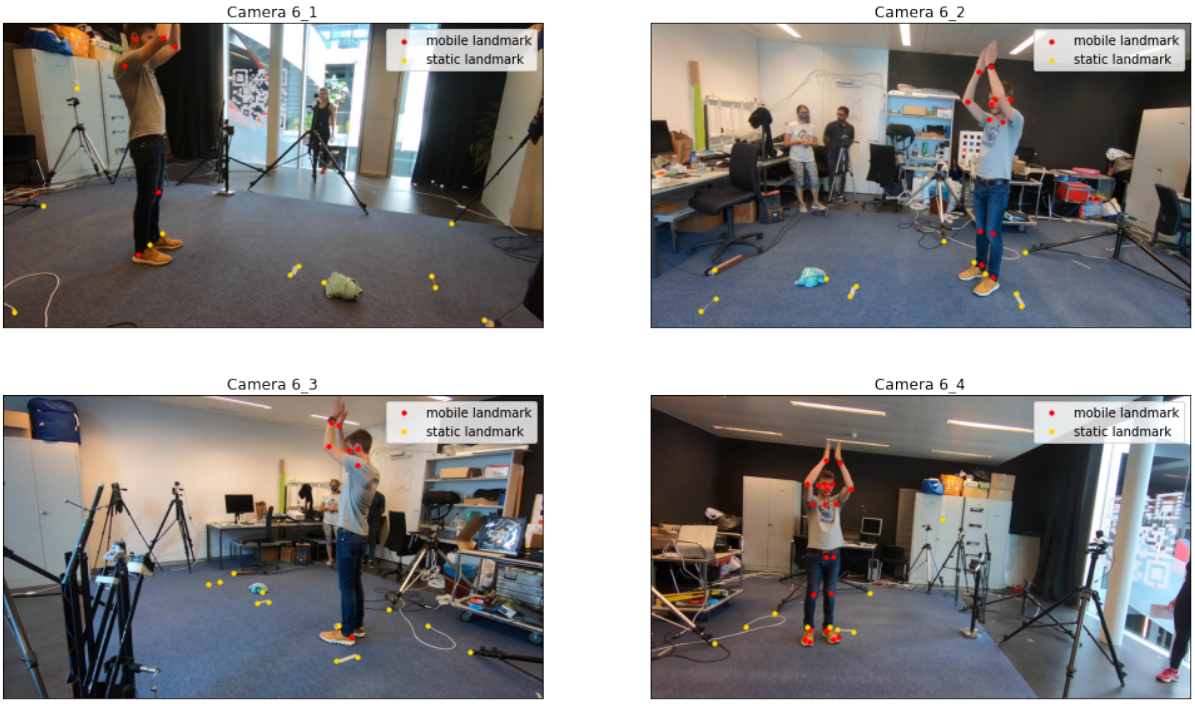


**Figure 8: Mobile and static landmarks.** Mobile (red) and static (yellow) landmarks in the first frame. Mobile landmarks have been obtained though OpenPose [4] 2D pose estimation whereas static landmarks have been manually selected.

After that camera intrinsics and extrinsics parameters are found, we are able to apply the 3D pose estimation algorithm detailed in section 4.1.3. Figure 9 shows the results of the 2D pose estimation done with OpenPose for the sample frames previously shown in figure 7. Uncertain joints (confidence $< 0.9$) are not displayed. Corresponding 3D poses are shown in figure 10.

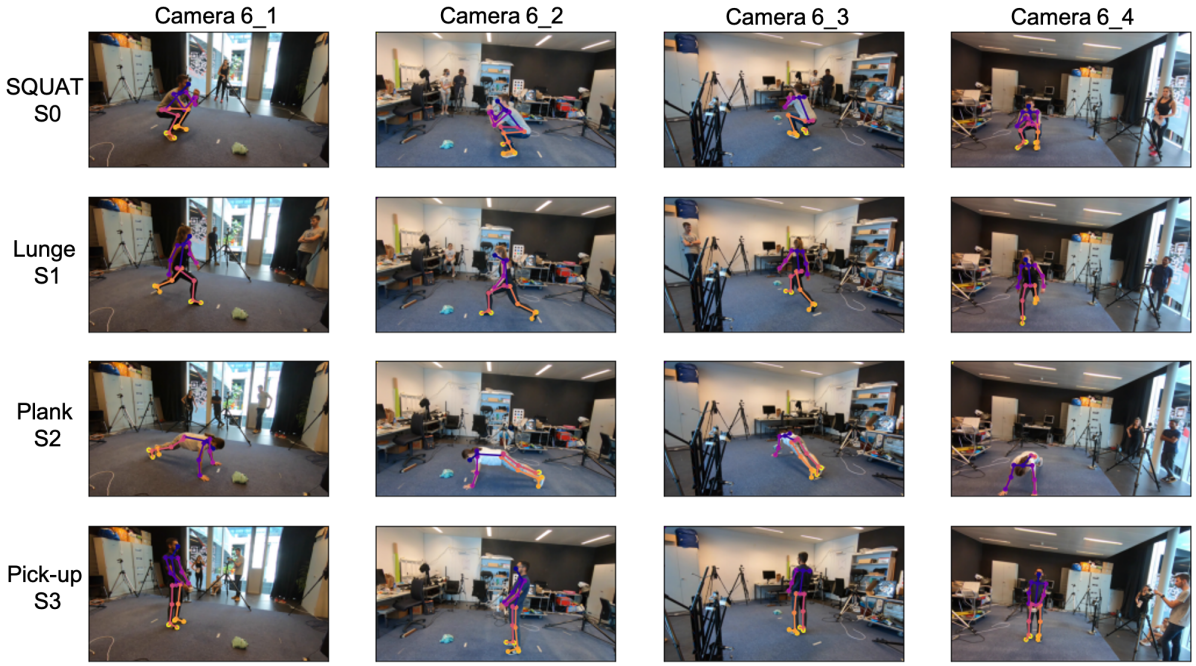


**Figure 9: Sample 2D pose estimations.** Same frames as in figure 7. Each row shows an action performed by a subject and viewed from the 4 different cameras. Body joints detected by OpenPose [4] and having a confidence score above 0.9 are displayed on top of the image. Joints are linked using the bone structure displayed in figure 3.

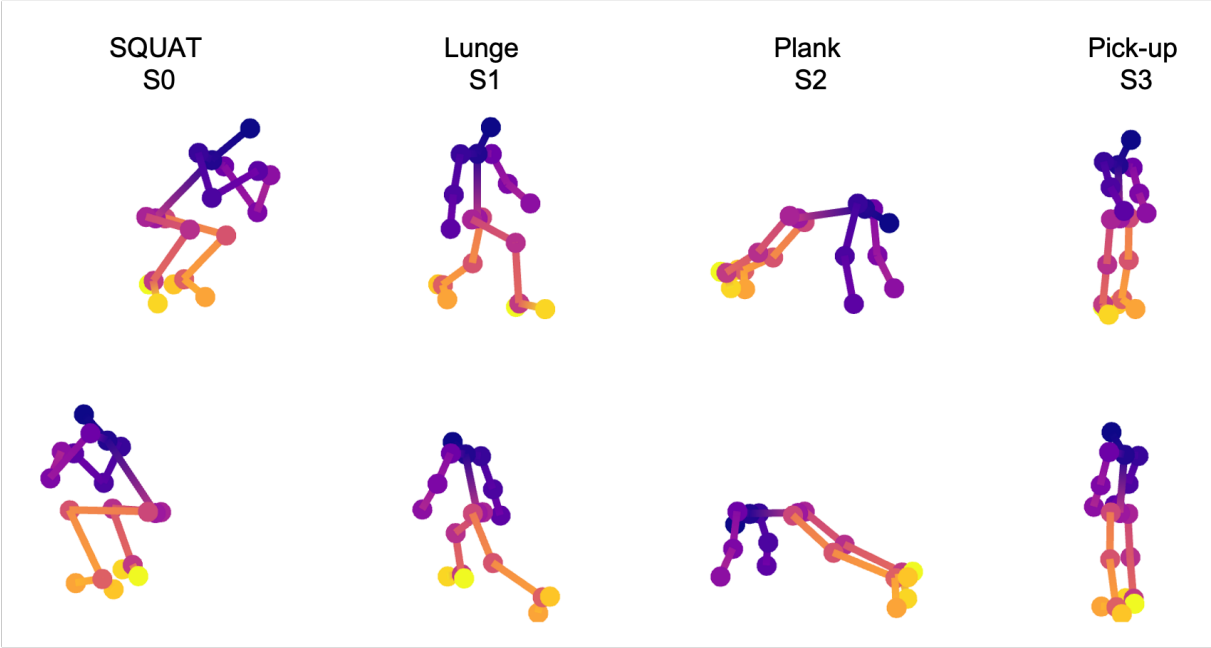


**Figure 10: Sample 3D pose estimations.** Estimated 3D poses obtained from the 2D poses displayed in figure 9. Top and bottom are the same poses observed from a different angle.

Once 3D pose is estimated, repetitions of the same movement are separated as explained in section 4.1.4. Examples of plots used for the identification of single movement are shown in figure 11. Table 3 summarize the number of repetitions extracted for each action, subject and instruction. The average length of a movement is 79, 100 and 58 frames for SQUATs, lunges and planks, respectively.

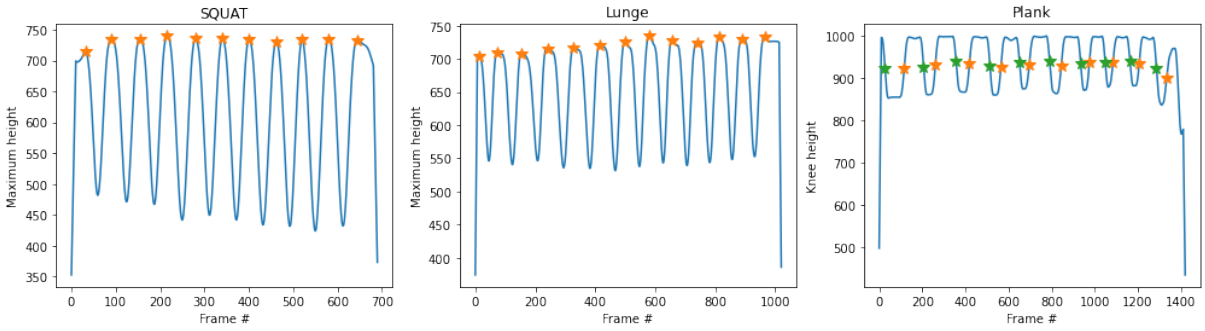


**Figure 11: Individual movements separation plots.** The fist two plots show the maximum vertical distance between joints evolution along frames (blue curve) for sequences of SQUATs and lunges. Single movements are separated by peaks (orange stars). The plot on the left shows the knee height along frames (blue curve) for a sequence of planks. Movement are taken between positive (in orange) and negative (in green) peaks of the first discrete difference.

## 5.2 Motion Correction

When all the movements have been labeled, sequences of 3D poses are fed to the motion correction model described in figure 4. Qualitative results of this model on the test subject (number 3) for every instruction are shown in figure 12.

## 5.3 Pipeline

If sequences of 3D poses are not already available, they need to be estimated from raw videos before being corrected by our model. Examples of such 3D pose estimation for one frame are shown in figure 13 in SMPL output format.

## 5.4 Action Classification

In order to validate and estimate the performance of our action correction model, we feed original movements and corrected ones to the classifier described in 5. The performance of this classifier on the original data for each action and instruction is described in the confusion matrix shown in table 4. Then Correction model performance can be assessed thanks to the classification results of the corrected movements visible in table 5.

| Action | Instruction | Subject | Repetitions | Total / instruction | Total / action |
|---|---|---|---|---|---|
| SQUATs | Correct | 0 | 10 | 41 | 132 |
| | | 1 | 10 | | |
| | | 2 | 11 | | |
| | | 3 | 10 | | |
| | Feet too wide | 0 | 5 | 23 | |
| | | 1 | 8 | | |
| | | 2 | 5 | | |
| | | 3 | 5 | | |
| | Knees inward | 0 | 6 | 23 | |
| | | 1 | 7 | | |
| | | 2 | 5 | | |
| | | 3 | 5 | | |
| | Not low enough | 0 | 5 | 21 | |
| | | 1 | 7 | | |
| | | 2 | 5 | | |
| | | 3 | 4 | | |
| | Front bent | 0 | 5 | 24 | |
| | | 1 | 6 | | |
| | | 2 | 6 | | |
| | | 3 | 7 | | |
| Lunges | Correct | 0 | 12 | 46 | 127 |
| | | 1 | 11 | | |
| | | 2 | 11 | | |
| | | 3 | 12 | | |
| | Not low enough | 0 | 10 | 40 | |
| | | 1 | 10 | | |
| | | 2 | 10 | | |
| | | 3 | 10 | | |
| | Knee passes toe | 0 | 10 | 41 | |
| | | 1 | 10 | | |
| | | 2 | 11 | | |
| | | 3 | 10 | | |
| Planks | Correct | 0 | 7 | 33 | 103 |
| | | 1 | 8 | | |
| | | 2 | 11 | | |
| | | 3 | 7 | | |
| | Arched back | 0 | 5 | 30 | |
| | | 1 | 5 | | |
| | | 2 | 11 | | |
| | | 3 | 9 | | |
| | Hunch back | 0 | 10 | 40 | |
| | | 1 | 10 | | |
| | | 2 | 11 | | |
| | | 3 | 9 | | |

**Table 3: Number of repetitions.**

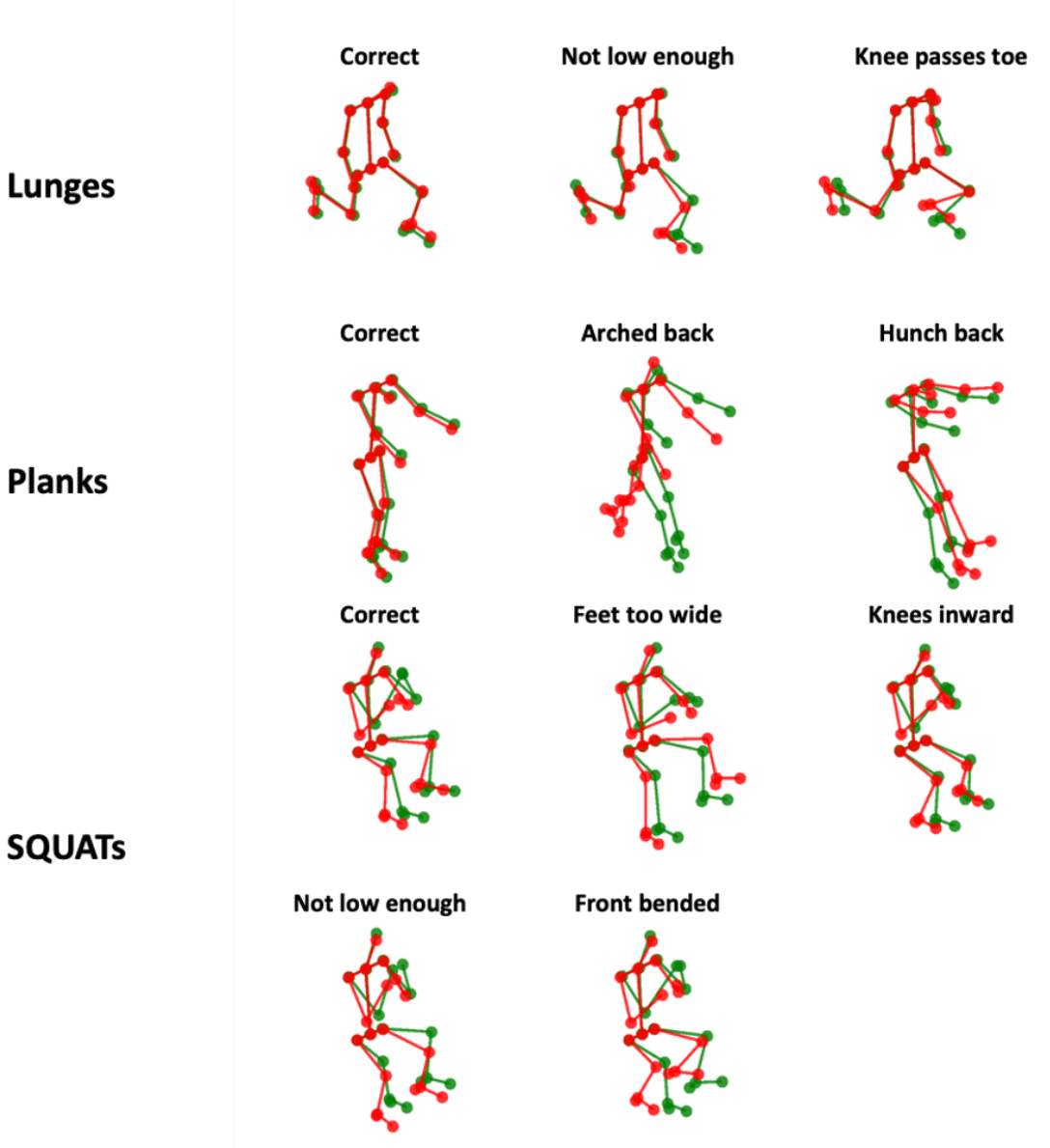


**Figure 12: Qualitative results of correction model.** The red skeleton represents the model input (original pose), whereas the green skeleton is the corrected pose. Planks are standing upright because of the skeleton normalization orienting the spine vertically.


**Figure 13: Sample outputs from VIBE.** Estimated 3D poses from VIBE algorithm [22].

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQUATs | Correct | **6** | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 60% | 89% |
| | Feet too wide | 0 | **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% | |
| | Knees inward | 0 | 0 | **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% | |
| | Not low enough | 0 | 0 | 0 | **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100% | |
| | Front bent | 0 | 0 | 1 | 0 | **6** | 0 | 0 | 0 | 0 | 0 | 0 | 86% | |
| Lunges | Correct | 0 | 0 | 0 | 0 | 0 | **10** | 0 | 2 | 0 | 0 | 0 | 83% | 94% |
| | Not low enough | 0 | 0 | 0 | 0 | 0 | 0 | **10** | 0 | 0 | 0 | 0 | 100% | |
| | Knee passes toe | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **10** | 0 | 0 | 0 | 100% | |
| Planks | Correct | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **7** | 0 | 0 | 100% | 100% |
| | Arched back | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **9** | 0 | 100% | |
| | Hunch back | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **9** | 100% | |

**Table 4: Confusion matrix of classification model.**

| Action | Instruction | Correct | Incorrect | Accurately corrected | |
|---|---|---|---|---|---|
| SQUATs | Correct | **10** | 0 | 100% | 97% |
| | Feet too wide | **5** | 0 | 100% | |
| | Knees inward | **5** | 0 | 100% | |
| | Not low enough | **4** | 0 | 100% | |
| | Front bent | **6** | 1 | 86% | |
| Lunges | Correct | **11** | 1 | 92% | 64% |
| | Not low enough | 1 | **10** | 10% | |
| | Knee passes toe | **9** | 1 | 90% | |
| Planks | Correct | **7** | 0 | 100% | 100% |
| | Arched back | **9** | 0 | 100% | |
| | Hunch back | **9** | 0 | 100% | |

**Table 5: Classification model results on corrected movements.** Incorrect class regroups all instructions that are not correct.

# 6 Discussion

The dataset generated in this project contains 363 individual movements. Each of these movements is defined by two 2D pose sequences obtained either by 2D pose estimation or by 3D pose re-projection on all four cameras and one 3D pose sequence. Even if this dataset is relatively small compared to other 3D poses datasets such as MPI-INF-3DHP [33], Human3.6M [17], 3DPW [41] or AMASS [28], its uniqueness stands in the fact that it contains different versions of the same movement done by the same subjects. This allows the comparison between these movements and also for the correction of incorrect movements. It is also worth mentioning that subjects had their own interpretation of the given instructions meaning that the correct action from one subject can differ from the one to another. If this inter-subject variability can help generalization of the algorithm on unseen 3D pose sequences, we have no guarantee that movement labelled as correct are exempt from any mistake. This aspect should be confirmed by specialists such as physiotherapists or movement specialists even if, as already exposed in the introduction, the new ecological approach of movement suggests that the perfect movement can differ depending on the environment and the individual.

We will now discuss the results of the correction model and especially the ones presented in figure 12, which are representative of the ones of all the testing set. In the first row we can see the correction applied to poses related to lunges: for the *correct* one there almost is not any correction, which is highly desirable. The two other on the other hand are corrected at the leg level. For the one originally *not low enough* the front knee is raised, which is equivalent to lower the hips since the skeleton is centered so the middle hip is at the center. The main correction made on the lunge where the *knee passes toe* is an augmentation of the angle between the front thigh and calf so the knee does not pass the toe anymore. For planks, the *correct* movement almost stays untouched, whereas for the incorrect ones the corrections are quite obvious. In *arched back* planks feet and knees are pushed down, which corresponds to raising the hips and arms are moved forward. Inverse modifications are applied to *hunch back* planks, which is logical since the two mistakes are the opposite one of another. Concerning the SQUATs, results are more blurry: for every pose, all the body joints except shoulders and hips present some correction. It is a bit unexpected that the correction model also gives strong correction to the arms since mistake categories specified in the instruction were focusing on the legs. However, if we take only the lower body into account, incorrect movements are corrected toward the right direction. Knees and feet are brought closer for the one with *feet too wide*, knees are pushed away for one with *knees inward*. For the two last ones the same kind of correction as in the incorrect lunges seems to apply, meaning that when the SQUAT is *not low enough* knees are pushed higher and when it is *front bent*, thigh-calf angles are increased. There is still some correction applied to the *correct* SQUAT legs that is a bit similar to the one not being low enough meaning that maybe the correct SQUAT of the test subject are not low enough either.

In general, one can say that qualitatively speaking, the corrections applied to incorrect movement are clearly going toward the direction of a correct one. However, these results can still be improved, in particular when considering the arm correction in SQUATs. An hypothesis to explain this could be that there are not enough examples of correct SQUATs to exhibit the full variability of the arm during the action (some are doing them with arms

bent and hands together while other do them with arms hold straight ahead of them).

The motion classifier allows us to have a more quantitative idea of the correction performance. But before being able to judge this, we need to assess the performance of the classifier itself on the data. This is what is shown in table 4. There, we can observe that the largest numbers of each rows are lying on the diagonal which already tells us that the classification is mainly correct. Actually, the classifier succeed to correctly classify all the planks as well as incorrect lunges and some incorrect SQUATs. One *front bent* SQUAT is classified as having *knees inward* and by looking at the video of this specific movement it is indeed not so clear because the subject almost fell. On the other hand, classification of *correct* SQUATs and lunges is not perfect, since they are exhibiting accuracies of 60% and 83%, respectively. Three *correct* SQUATs are classified as *not low enough* while one is classified as *front bent*. This correlates with the observation made above, that is that the *correct* SQUAT shown seems to be corrected as if it is *not low enough*. Concerning *correct* lunges, 2 are classified as *knee passes toe*. Keeping this classification performance in mind, we are now able to use this classifier on corrected movements and as visible in table 5. Here, 97% of SQUATs and 100% of planks are classified as correct after correction, meaning that this correction is indeed done accurately. For lunges however, if correct and knee pass toe are almost all accurately corrected, there is only one *not low enough* that appears to be classified as correct after the correction. Since qualitative results of this correction (figure 12 top middle) does not seem aberrant, an hypothesis can be that even if the correction goes toward the right direction, it is not strong enough for the movement to be classified as *correct* afterwards.

Overall, this work has proven that it is possible to train a model so that it gives personalized corrections to be made on a sequence of poses in order to improve its realization. Even if there are still improvements to be made, we believe that training this model on a larger dataset containing a larger variety of subjects and mistakes can significantly improve its performance. Next steps also include making the whole pipeline work live. If for now implemented pipeline with VIBE 3D pose estimator and our correction model is able to work at a speed higher that 15fps, the correction model is only trained to work on full sequences meaning that it can only provide feedback once the movement is completely done. Further work can thus focus on giving correction to incomplete movement based on previous estimated poses, this could be for example coupled with motion prediction where predicted future poses would be corrected and given to the subject as target poses to reach.

# 7 Acknowledgements

I want to strongly thank the CVLAB for allowing me to do this project in such stimulating environment and more particularly my two supervisors Sena Kiciroglu and Isinsu Katircioglu for their always useful feedback and discussions.

# 8 References


[1] Talal Alatiah and Chen Chen. Recognizing Exercises and Counting Repetitions in Real Time. *arXiv*, pages 1–13, 2020.

[2] Julie Barlow, Chris Wright, Janice Sheasby, Andy Turner, and Jenny Hainsworth. Self-management approaches for people with chronic conditions: A review. *Patient Education and Counseling*, 48(2):177–187, 2002.

[3] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.

[4] Zhe Cao, Tomas Simon, Shih En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua(Xxx):1302–1310, 2017.

[5] João Carreira and Andrew Zisserman. Quo Vadis, action recognition? A new model and the kinetics dataset. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:4724–4733, 2017.

[6] Steven Chen and Richard R. Yang. Pose Trainer: Correcting Exercise Posture using Pose Estimation. *arXiv*, 2020.

[7] Marco Cuturi and Mathieu Blondel. Soft-DTW: A differentiable loss function for time-series. *34th International Conference on Machine Learning, ICML 2017*, 2:1483–1505, 2017.

[8] M. Esat Kalfaoglu, Sinan Kalkan, and A. Aydin Alatan. Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition. *arXiv*, pages 1–19, 2020.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. *arXiv*, 2018.

[10] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:4346–4354, 2015.

[11] Andrew C. Fry, J. Chadwick Smith, and Brian K. Schilling. Effect of Knee Position on Hip and Knee Torques during the Barbell Squat. *Journal of Strength and Conditioning Research*, 17(4):629–633, 2003.

[12] Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. SMART Frame Selection for Action Recognition. 2020.

[13] Liang-yan Gui, Yu-xiong Wang, Xiaodan Liang, and M F Moura. Adversarial Geometry-Aware Human Motion Prediction. *European Conference on Computer Vision (EECV)*, page 786–803, 2018.

[14] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[15] De An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (ii):7366–7375, 2018.

[16] Nathan Hutting, Venerina Johnston, J. Bart Staal, and Yvonne F. Heerkens. Promoting the use of self-management strategies for people with persistent musculoskeletal disorders: The role of physical therapists. *Journal of Orthopaedic and Sports Physical Therapy*, 49(4):212–215, 2019.

[17] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6M. *Ieee Transactions on Pattern Analysis and Machine in[U+2121]Ligence*, page 1, 2014.

[18] Ashesh Jain, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:5308–5317, 2016.

[19] Venerina Johnston, Gwendolen Jull, Dianne M. Sheppard, and Niki Ellis. Applying principles of self-management to facilitate workers to return to or remain at work with a chronic musculoskeletal condition. *Manual Therapy*, 18(4):274–280, 2013.

[20] Zachary Kerr, Christy Collins, and Rachel Dawn. Epidemiology of Weight Training-Related Injuries Presenting to United States Emergency Departments, 1990 to 2007. *The American journal of sports medicine*, 38:765–771, 2010.

[21] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(4), 12 2018.

[22] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 5252–5262, 2020.

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2556–2563, 2011.

[24] Jeremy Lewis and Peter O'Sullivan. Is it time to reframe how we care for people with non-traumatic musculoskeletal pain? *British Journal of Sports Medicine*, 52(24):1543–1544, 2018.

[25] Yinxiao Li, Zhichao Lu, Xuehan Xiong, and Jonathan Huang. Perf-net: Pose empowered rgb-flow net. *arXiv*, 2020.

[26] An-lun Liu. A Posture Evaluation System for Fitness Videos based on Recurrent Neural Network.

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6), 2015.

[28] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael Black. AMASS: Archive of motion capture as surface shapes. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:5441–5450, 2019.

[29] John Makhoul. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):27–34, 1980.

[30] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. *arXiv*, 2020.

[31] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:9488–9496, 2019.

[32] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:4674–4683, 2017.

[33] 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. *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, pages 506–516, 2018.

[34] Christopher J.L. Murray. The State of US health, 1990-2010: Burden of diseases, injuries, and risk factors. *JAMA - Journal of the American Medical Association*, 310(6):591–608, 2013.

[35] Samsu Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan. Human action recognition using Dynamic Time Warping. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011*, (July):1–5, 2011.

[36] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only Time Can Tell: Discovering temporal data for temporal modeling. *arXiv*, 2019.

[37] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. (November), 2012.

[38] Atima Tharatipyakul, Kenny Choo, and Simon T. Perrault. Pose Estimation for Facilitating Movement Learning from Online Videos. *arXiv*, (1):1–5, 2020.

[39] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann Lecun, and Manohar Paluri. A Closer Look at Spatiotemporal Convolutions for Action Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[40] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. *Bundle Adjustment — A Modern Synthesis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[41] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11214 LNCS:614–631, 2018.

[42] Borui Wang, Ehsan Adeli, Hsu Kuang Chiu, De An Huang, and Juan Carlos Niebles. Imitation learning for human pose prediction. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob(2):7123–7132, 2019.

[43] Haoran Xie, Atsushi Watatani, and Kazunori Miyata. Visual feedback for core training with 3D human shape and pose. *Proceedings - 2019 NICOGRAPH International, NicoInt 2019*, pages 49–56, 2019.

[44] Feng Zhou and Fernando De La Torre. Generalized time warping for multi-modal alignment of human motion. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, 2012.