AR mirror for sportive training application

1st Alessandro Castellaz
Department of Industrial Engineering
University of Trento
alessandro.castellaz@studenti.unitn.it

2nd Francesco Mazzoni
Department of Industrial Engineering
University of Trento
francesco.mazzoni@studenti.unitn.it

***Abstract*—This paper provides an analysis focused on the use of the library MediaPipe for sportive training purposes. The project is based on a simple GUI that permits to choose between two different kind of exercises: biceps exercise and squat exercise.**

I. Introduction

Nowadays, with the development of new technologies, it's possible to implement the use of machine learning in order to improve the quality of the sportive training. In this project the crucial element is the library MediaPipe Pose which allows the tracking and the analysis of the movements of the human body. Human pose estimation from video plays a critical role in various applications such as quantifying physical exercises, sign language recognition, and full-body gesture control. For example, it can form the basis for yoga, dance, and fitness applications. It can also enable the overlay of digital content and information on top of the physical world in augmented reality.

II. Basics of MediaPipe

MediaPipe Pose is a Machine Learning solution for high-fidelity body pose tracking, inferring 33 3D landmarks (as shown in Figure 1) and background segmentation mask on the whole body from RGB video frames utilizing our BlazePose [1] research that also powers the ML Kit Pose Detection API. The solution utilizes a two-step detector-tracker ML pipeline. Using a detector, the pipeline first locates the person/pose region-of-interest (ROI) within the frame. The tracker subsequently predicts the pose landmarks and segmentation mask within the ROI using the ROI-cropped frame as input. For video use cases the detector is invoked only as needed, i.e., for the very first frame and when the tracker could no longer identify body pose presence in the previous frame. For other frames the pipeline simply derives the ROI from the previous frame's pose landmarks. The detector BlazePose is inspired by the lightweight BlazeFace model in Media Pipe, used in MediaPipe Face Detection, as a proxy for a person detector. It explicitly predicts two additional virtual keypoints that firmly describe the human body center, rotation and scale as a circle. Inspired by Leonardo's Vitruvian man, the algorithm predicts the midpoint of a person's hips, the radius of a circle circumscribing the whole person, and the incline angle of the line connecting the shoulder and hip midpoints. The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks, that are than used in order to keep track of the person's specific motion. The output of the algorithm is a list of pose landmarks. Each landmark consists of the following:

- x and y coordinates: landmark coordinates normalized to [0.0, 1.0] by the image width and height respectively.
- z coordinate: represents the landmark depth with the depth at the midpoint of hips being the origin, and the smaller the value the closer the landmark is to the camera. The magnitude of z uses roughly the same scale as x.
- visibility: a value in [0.0, 1.0] indicating the likelihood of the landmark being visible (present and not occluded) in the image.

There is also a variation of the pose landmarks, called ”POSE WORLD LANDMARKS” that provides the set of coordinates in the 3D world. Also in this case the value of the visibility is provided. The last type of output, i.e. ”SEGMENTATION MASK” is not taken into consideration since it was not used in our implementation.

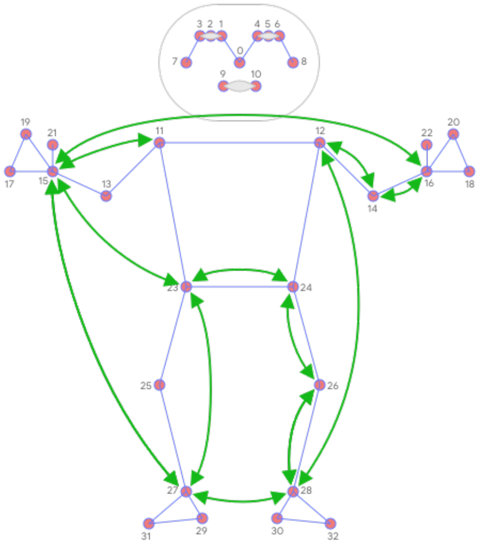

Fig. 1. Pose landmarks

III. Pose estimation problem and case study

Pose Estimation is a general problem in Computer Vision where the goal is to detect the position and orientation of a person or an object. Usually, this is done by predicting the location of specific keypoints like hands, head, elbows, etc.

in case of Human Pose Estimation. Possible applications of this type of problem regard the movement or position of a person in space and check if they are correct wrt a reference. In particular it is common that a person checks the correct motion of arts during sport activities either with the help of a trainer or by itself. A human pose estimator can do the same job and help the user correct its movements. This is what we propose in our project. The user will choose between two training exercises (i.e. biceps training and squat training) and he will try to complete the requested tasks. Meanwhile the human pose estimator will analyze how the exercises are performed, being able to count the repetitions and to see where the keypoints are located. The human pose estimator will also provide a feedback on how well the exercise is performed, by monitoring the keypoints relative position wrt a reference. In the following subsections we will explain the two different exercises in terms of logic, major commands and procedures. For a more detailed explanation, see the Media Pose documentation and our Readme.md file.

A. The biceps exercise

The first exercise the user can choose is about arm bending and biceps training. The user is asked to perform a fixed number of repetitions (i.e. 5 times) for each arm, starting with both arms along its body. A repetition will occur if the angle between the shoulder, the elbow and the wrist (namely the corresponding keypoints) will be below 30 deg and if the arm comes from a ”down” position (meaning that the keypoints form an angle greater than 150 deg). The exercise is completed as the number of repetitions will be reached for both arms. The angle is calculated by simply applying the following function:

$$\theta = arctan\Big(\frac{|y_C - y_B|}{|x_C - x_B|}\Big) - arctan\Big(\frac{|y_A - y_B|}{|x_A - x_B|}\Big) \quad (1)$$

where x_i and y_i are the x and y coordinates of the keypoints of interest, being $i \in \{A, B, C\}$. Notice that the angle is expressed in radiants, thus a conversion in deg has to be performed if needed. The angle has a maximum value of 180 deg, thus every time this maximum value is exceeded the angle is corrected as follows: $\theta = 360 - \theta$. As the user shows himself in front of a webcam and performs the exercise, the human pose estimator will detect also the position of the top part of the arm wrt the body. We expect that a good biceps exercise is performed if the angle between elbow, shoulder and hip (defined as keypoints) is less than 20 deg. Thus, the human pose estimator will check the correct position of the arm at each frame. For what regards the logic of the algorithm, initially both arms start lying along the user’s body. In this case the position is marked as ”down”. As the hand is raised towards the shoulder and reaches the threshold (30 deg), the repetition counter is increased and the position is marked as ”up”. As the hand goes towards the starting position, the angle will be higher than the second threshold (150 deg, different than before to avoid chattering effects) and the position will be marked again as ”down”. As both left-arm- and right-arm counter reach the target repetition value, the algorithm stops.

B. The squat exercise

In this exercise the user is asked to perform 10 squat repetitions. A repetition will occur every time the user, starting from a standing position, bends its knees forming an angle below 115 deg with his hips, knees and ankles. The angle calculated as follows:

$$\begin{cases} \theta_L = arctan\Big(\frac{|y_C^L - y_B^L|}{|x_C^L - x_B^L|}\Big) - arctan\Big(\frac{|y_A^L - y_B^L|}{|x_A^L - x_B^L|}\Big) \\ \theta_R = arctan\Big(\frac{|y_C^R - y_B^R|}{|x_C^L - x_B^L|}\Big) - arctan\Big(\frac{|y_A^R - y_B^R|}{|x_A^R - x_B^R|}\Big) \\ \theta = min(\theta_L, \theta_R) \end{cases} \quad (2)$$

where y_i^j is the y coordinate of the i-th keypoint on the j-th side, being $i \in \{A, B, C\}$ and $j \in \{L, R\}$. The keypoints of interest are the one corresponding to the hips, knees and ankles. As the user shows itself in front of the webcam and performs the exercise, the human pose estimator will detect also the position of the knee wrt the ankle and the foot index, both on the left part and on the right part of the user body. We expect that a good squat exercise is performed if the position of the knee does not exceed too much the area of the corresponding foot on the same leg. In other words, the angle between knee, ankle and foot index must be higher than 115 deg. Thus every time the position of the knee exceeds the limit of 115 deg the program will warn the user about his position. The angles are calculated as shown in equation (2) but with different keypoints. Finally, it is also checked how down the user’s pelvis goes, by tracking the estimation of the distance of the hip from the heel. In order to achieve this result, the ”pose world landmarks” are used and knee and heel positions on both sides are extracted. In this way we can keep track of the estimation of the pelvis from ground and show it to the user in order to provide a physical information. The pelvis height estimation is obtained as follows:

$$\begin{cases} h_{hL} = |y_{hip}^L - y_{heel}^L| \\ h_{hR} = |y_{hip}^R - y_{heel}^R| \\ h = \frac{h_{hL} + h_{hR}}{2} \end{cases} \quad (3)$$

The mean value allows to estimate the height in the middle of the two distances, i.e. where the pelvis should be located, as the user may be oriented with different angles wrt the camera. Moreover, it is usefull to check when the squat is deep enough and inform the user with an encouraging sentence. For this purpose, the relative height between knees and hips is considered. If this value stays below 0.025, than the user has gone down enough and the exercise is performed correctly from this point of view. The relative height is calculated as follows:

$$\begin{cases} h_L = |y_1^L - y_2^L| \\ h_R = |y_1^R - y_2^R| \\ d = min(h_L, h_R) \end{cases} \quad (4)$$

where y_i^j is the y coordinate of the i-th keypoint on the j-th side, being $i \in \{1, 2\}$ and $j \in \{L, R\}$. The keypoints are the ones of the hip and of the knee. Notice that we consider the minimum values of angles and distances used for detection purposes due to the fact that the user may position himself with one leg closer to the camera, either the right one or the left one. In a tilted position, the angles may differ one from the other. For what concerns the algorithm logic we have that the user body starts in a classic standing position and it will be marked as ”up”. As he bends his knees below 115 deg, the counter will increase and the position will be marked as ”down”. As the user will increase again the angle between hip, knee and ankle above 150 deg (greater than before to avoid possible chattering) the user position is marked again as ”up” and it will be ready for the detection of the next repetition. The algorithm works as the repetition counter reaches the desired value.

IV. Conclusions

The algorithm we tested shows good results in terms of tracking of the features and angles calculations. These results are achieved as the user stands in front of a camera in a proper way and the illumination is quite high as well as constant. The algorithm for the biceps exercise is able to manage the keypoints and elaborate useful results since the two arms are easily seen by the webcam. Indeed, no occlusions are present in the scene as far as the user stands in front of the camera with no relative rotation. For what concerns the squat exercise instead results are quite good in terms of ”up” or ”down” position detection, but there are some issues in the parts regarding the error detection. This is due to the position of the landmarks wrt the webcam, that has to be oriented properly as the exercise starts. Namely, issues may occur be due to the rotation of the user wrt the camera. If the user is perfectly parallel, the correction angle (knee, ankle and foot index) is higher than the one obtained with a tilted position. In this case a warning is given if a big knee-bending occurs. On the other hand, if the user is slightly tilted, the correction angle is closer to the real movement and it is better appreciated as well as the knee-bending. A warning will also occur more easily since the value estimated is lower than the previous case and consequently closer to the threshold. The drawback is that the leg closer to the camera partially occludes the further one, causing some issues with the robustness of the landmarks. Thus a good tradeoff position has to be found. Generally speaking, we can say that the results obtained are good enough to let the code be used for training purposes. All warnings work quite good in order to provide some useful feedback in case the movement must be corrected.

References

[1] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, Matthias Grundmann, “BlazePose: On-device Real-time Body Pose tracking,”, June 2020.