Sport Tech
- TraBasT (TRAcked BASketball Training) -

Edoardo Fiorentino [248444] Carlo Serra [247189]

1 Introduction

In this report we discuss **TraBasT (TRAcked BASketball Training)**, a Sport Tech course project that uses motion capture to track basketball players movement during a free throw. The final goal is to help amateur players learn and improve their free throws by comparing their free throw motions against a gold standard and giving them performance based feedback. The project source code is available on GitHub[1].

2 What is mocap

Motion capture[2] technology is a sophisticated method used to record the positions of objects or body parts in space over time. The recording process converts this information into digital data, which can be further analyzed, used to produce animations, or integrated with other applications. It is widely employed in diverse fields such as entertainment, sports, and medical research.

2.1 Optical Marker-Based Motion Capture

Motion capture serves as the foundation for various systems that adapt its core concept to suit their specific needs, utilizing different technologies depending on their intended applications. The data utilized in the initial task of the assignment were generated using an Optical Marker-Based Motion Capture system[3]. This technique, perhaps the most prevalent and effective, employs a set of high-speed cameras to track the movement of objects in three-dimensional space via reflective markers placed on key points of the object.

For our goal, we exploited the technology provided by Qualisys[4], since we needed a portable and flexible setup to use in Sanbapolis. Our setup was composed by eight cameras.

3 WorkFlow organization

We subdivided our work in different and clear activities:

- Learn computer vision skills (OpenCV, Open3D, Numpy)
- Record videos using MoCap in Sanbapolis tracking the players and the ball
- Load and manage the data
- Analyze and evaluate performances (User against Gold Standard)
- Define recommendations depending on the player movements
- Test

Then we set three milestones to reach:

- Data Collection, Data Conversion, Data Reading
- Performance Evaluation
- Feedback Integration

4 Our work in practice

4.1 Data Collection and Conversion

We collected the necessary data in Sanbapolis' gym with MoCap, being some of the shooters ourselves, using the Qualisys motion capture system [4]. For each shooter (four) we recorded ten good free throws and ten bad ones. In this way we compensated for the low number of players, gathering more data, and we also had references for very bad shots, making it easier to evaluate the overall performances.

In recording the free throws with MoCap we used the "Animation" setup of Qualysis, that requires 24 markers distribuited in specific position around the body to capture it during the motion. The 24 markers are then processed by the Qualysis software, and they are represented by the 24 labels in the CSV files. The 24 labels are:

1. Hips
2. Spine
3. Spine1
4. Spine2
5. Neck
6. Head
7. LeftShoulder
8. LeftArm
9. LeftForeArm
10. LeftForeArmRoll
11. LeftHand
12. RightShoulder
13. RightArm
14. RightForeArm
15. RightForeArmRoll
16. RightHand
17. LeftUpLeg
18. LeftLeg
19. LeftFoot
20. LeftToeBase
21. RightUpLeg
22. RightLeg
23. RightFoot
24. RightToeBase

To use the recordings, we preprocessed the data provided by Qualisys system into a practical format: the software did not allow us to automatically convert the data into a CSV format, so, having the files in a .mat (matlab) format, we wrote a script in matlab to convert them into CSV files.

4.2 Data Reading and Data Plotting

We developed a custom solution in Python to read the CSV files from which we extracted the x, y, and z coordinates of each joint of the skeletons for each frame. After converting and reading the data, we connected the joints of the skeleton and plotted it in 3D using both the MatPlotLib [5] and Open3D libraries [6]. Comparing the two options, we chose the second as the default because it provided better visual outcomes. Since it required some time to get the data, we started working using data of a project where they tracked squat movements using mocap [7]. Due to occlusion, the data had missing points, so we implemented the Kalman Filter [8] to increase the quality of our data, but we found out later that we did not need it for our frames: the data we used were smooth and without any missing joint for any frame.

Occlusion Occlusion is a frequent issue in optical marker-based motion capture, occurring when cameras lose sight of markers or their reflections fluctuate due to environmental factors, such as floor reflections in Sanbapolis gym. This results in data gaps, inaccuracies in motion tracking, and a flickering effect when the data are plotted.

Kalman Filter The Kalman Filter is a widely used algorithm in robotics and computer vision, predicting and correcting system states iteratively. It excels at handling noisy measurements and uncertainties, incorporating data like velocity and acceleration for better estimates. We used the implementation from OpenCV, assuming uniformly accelerated motion.

We crafted the data reading and data plotting code starting from a computer vision project about motion capture [9].

4.3 Performance Evaluation

To evaluate free throw performance, we compare each movement to a gold standard. By analyzing the differences between the two, we can assess the quality of the shot and identify areas for improvement.

4.3.1 Preparing the skeletons to the analysis

As a first step, we have to compare the free throw of the player to another free throw, which serves as the gold standard. To achieve this, we superpose the two moving skeletons performing the free throws. This allows us to compute the distances between corresponding joints in each frame and analyze the differences in movement between a regular free throw and the gold standard. The code for this is located in the *shoot_analysis.py* file.

Scaling the skeletons Every player has different heights and anatomical proportions. To evaluate the shot, we compare two different players (*user* and *gold standard*). For a general evaluation, it is necessary to normalize the size of the skeletons. To achieve this, we defined a standard cumulative joint length (the sum of the lengths of all joints in one skeleton), referred to as *target_joints_length* in the code. The ratio between the target length and the total length of the skeleton to be scaled (*current_joints_length*) is computed and then applied to scale the points of each frame in the CSV file (skeleton). This process is repeated for every frame.

$$scaled\_skeleton = \frac{target\_joints\_length}{current\_joints\_length} \times skeleton$$

Superposing the Skeletons To compute the differences, the skeletons must be aligned to a fixed reference point. This is necessary because each player starts from a different position and may move during the shot, which can lead to inaccurate measurements. We chose the pelvis as the reference point because it is the most central and stable part of the body. Fixing the pelvis, we can ensure consistency in the distances and alignment of the joint positions between the two skeletons during the analysis.

To align the skeletons, we calculated the displacement between the pelvis of the user and that of the gold standard. We then applied this displacement to the second skeleton to superpose them.

Video Cutting and Downsampling To compare the two free throws, two conditions must be met: players must start their shots at the same time, and the videos must have the same length. This ensures that the complete movement can be compared from start to finish.

After aligning the pelvises and scaling the dimensions, we manually trimmed the videos and downsampled the longer one. The downsampling was done by selecting the longer video and using the *numpy.linspace()* function to extract equally spaced frames. This resulted in two videos of equal length.

4.3.2 Evaluation Metrics

To ensure an accurate and meaningful assessment of free throws, we developed a set of evaluation metrics that measure the similarity between the player's movements and the gold standard.

We assigned an overall score to the movement, which is composed of several components: scores for the movement correctness of three body regions (arms, legs, and others) and a score for the speed of the free throw.

Metrics The scores for the arms, legs, and other regions are calculated as follows: for each frame and for each body part, we compute the distance between the corresponding joint points of the user and the gold standard.

For each point (x, y, z), for each frame, the euclidean distance between player and gold standard is calculated:

$$distance_i = \|\mathbf{points\_player}_i - \mathbf{points\_GS}_i\|_2 = \sqrt{\sum_{j=1}^{n} (points\_player_{ij} - points\_GS_{ij})^2}$$

where *i=(1, ..., 24)* represents the different point of the body.

To provide a more comprehensive evaluation, we also include angles in the scoring. For each part of the body, at least one angle is calculated: the elbow and arm angles (the angle formed between the arms and the body) for the arms, the knee angle for the legs, and the pelvis angle for the other parts. These angles are computed for each frame, but only the minimum or maximum value is considered for each angle, depending on the purpose of each one. Since the videos begin when the shooting motion starts, we can assume that the players are not in unusual positions but are actively moving to take the shot. By considering the minimum angles, we capture the bending of the elbows and knees in the gold standard, and we also know the initial position of the arms at the start of the movement (assuming that the minimum angle for the arms occurs at the beginning of the motion). Each player's angle is then compared to the corresponding gold standard angle.

To compute the angle between two segments, represented by the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, we use the following formula:

$$\theta = \arccos\left(\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}\right)$$

The resulting angle θ is in radians. To convert it to degrees, we use the formula:

$$angle\_in\_degrees = \frac{180}{\pi} \cdot \theta$$

Finally, to consider the internal angle, it is calculated as:

$$\theta_{internal} = 180^\circ - angle\_in\_degrees$$

Next, for each body part (arms, legs, other), we sum all the distances, add relative differences in angles between the player and the gold standard, and multiply the result by a coefficient. This coefficient, which was empirically determined, varies for each body part: it is high for the arms, as they are the most important, medium for the legs, which are still significant, and low for the other parts (such as the head, chest, or pelvis), since they have less influence on the overall movement.

The values of the coefficients are as follows:

- *ARMS_COEFF = 1.4*
- *LEGS_COEFF = 1.1*
- *OTHER_COEFF = 0.5*
- *SPEED_COEFF = 3*

All these metrics are then summed together, including the metric related to speed, as shown in the following.

$$\begin{aligned} OVERALL\_METRIC = & ARMS\_METRIC + LEGS\_METRIC + \\ & OTHER\_METRIC + SPEED\_METRIC \end{aligned}$$

To compute the metric on the arms:

$$ARMS\_METRIC = (\sum_{i=1}^{n} arms\_distances_i + arms\_angles\_metric) \times ARMS\_COEFF$$

where

$$\begin{aligned} arms\_angles\_metric = & |elbows\_R\_min\_diff| + |elbows\_L\_min\_diff| + \\ & |arms\_mean\_diff\_min| + |arms\_mean\_diff\_max| \end{aligned}$$

Mean of the differences of the angles (left and right) between player and gold standard:

$$angle\_mean\_diff = \frac{1}{n}\sum_{i=1}^{n} angle\_diff_i$$

$$angle\_diff_i = angle\_GS_i - angle\_player_i$$

For the arms mean, the formula is:

$$arms\_mean\_diff = \frac{arm\_R\_diff + arm\_L\_diff}{2}$$

To compute the metric on the legs:

$$LEGS\_METRIC = (\sum_{i=1}^{n} legs\_distances_i + |knees\_mean\_diff|) \times LEGS\_COEFF$$

Formula to calculate the *OTHER_METRIC* value:

$$OTHER\_METRIC = (\sum_{i=1}^{n} other\_distances_i + |pelvis\_mean\_diff\_min|) \times OTHER\_COEFF$$

The mean of the differences in pelvis angles (*pelvis_mean_diff_min*) is calculated in the same manner as for the arms. To calculate the pelvis angle, the angles of the right and left hip are calculated respectively. Specifically, the right hip angle is determined by the intersection of the segments connecting the Spine to the RightUpLeg, and the RightUpLeg to the RightLeg point. The same process is applied to calculate the left hip angle, using the Spine to LeftUpLeg and LeftUpLeg to LeftLeg segments.

The metric about the speed is computed as follows:

$$SPEED\_METRIC = |shot\_length\_player - shot\_length\_GS| \times SPEED\_COEFF$$

The two performance are compared in sense of length of the video, so is calculated the difference between the player and the gold standard, and then it is multiplied by a coefficient.

We also computed the range between the maximum and the relative minimum for arms angles: in specific we calculated the range of the right and left arms for the player and for the gold standard.

For both the player and for the gold standard, the range computed is the difference between the absolute maximum and the absolute minimum angles reached by them during the entire shot, from the start to the end. Considering i = *(right, left)*:

$$range\_angle_i = max\_angle_i - min\_angle_i$$

So we have 4 ranges: *range_R_arm, range_L_arm, range_R_arm_GS, range_L_arm_GS.*

All of these metrics (overall, arms, legs, and others) and their respective ranges are compared to threshold values to evaluate the performances of the free throws.

Each threshold is assigned an importance value ranging from 1 to 5, depending on the area being evaluated. However, the overall metric is given a higher importance value of 10, as it represents a comprehensive assessment of the shot. When a threshold is met, the corresponding importance value is added to a score parameter. By summing the importance values based on performance, we calculate the final score, which is then converted into a percentage to provide immediate feedback on the shot. This process essentially measures the quality of the shot, indicating how close it is to an optimal or ideal value.

$$percentage = \frac{\sum_{i=1}^{n} importance_i}{total\_importance} \times 100$$

Based on the calculated percentage, the performance is categorized as follows:

- **0% - 40%:** Beginner level – "You're a beginner. Focus on the basics."
- **41% - 60%:** Amateur player – "The basics are set, now refine your technique."
- **61% - 80%:** Good player – "Solid technique! Just fine-tune your shot."
- **81% - 90%:** Pro player – "Excellent! You're almost at the top."
- **91% - 97%:** GOAT player – "Incredible! Your performance is approaching the Gold Standard."
- **98% - 100%:** GOLD STANDARD – "Unbelievable! Your form matches the Gold Standard."

4.4 Feedback Integration

We integrated actionable recommendations for the user to enhance their performance based on the evaluation metrics we developed. The goal of the feedback system is to provide players with clear, personalized guidance that highlights areas of strength while pinpointing aspects of their shot that require improvement. By aligning the feedback with the specific metrics calculated from the player's movements, we ensure that the suggestions are both relevant and effective. This feedback not only supports the player in refining their technique but also motivates continuous improvement through targeted, easy-to-understand advice.

The feedback process starts with an evaluation of the primary general metrics - *overall*, *arms*, *legs*, and *other*. Each of these metrics is compared against a predefined threshold, where a lower value (below the threshold) indicates better performance. Recommendations are only provided for the *arms*, *legs*, and *other* metrics. If the performance in any of these exceeds the threshold (i.e., a higher score), the system will offer feedback on the specific areas that need improvement. For the *overall* metric, the feedback simply indicates whether the shot is good or if adjustments are needed, without diving into specific details.

If the *arms*, *legs*, or *other* metrics are below the threshold, it means the player's movements are well-aligned with the gold standard, and no specific feedback will be provided for these areas. If any of these metrics exceed the threshold, however, it indicates a significant discrepancy, and more detailed feedback will be given. For instance, if the *arms* metric exceeds the threshold, the feedback will examine the *arms_angles_metric*, analyzing discrepancies in the right and left elbow angles, the mean angle difference, and the range of motion between the player and the gold standard. Feedback might then recommend correcting elbow bending during the shot or increasing the range of arm motion to match the gold standard.

Similarly, when the *legs* metric is evaluated, the system examines elements such as *knees_mean_diff*, providing guidance on improving knee bending during the shot for greater stability and accuracy. For the *other* metric, feedback may focus on discrepancies in the *pelvis_mean_diff_min*, offering tips on body alignment and balance.

This structured approach ensures that the feedback is progressive and specific, starting from general advice and gradually narrowing down to more targeted recommendations. By ensuring that only areas where performance exceeds the threshold receive detailed feedback, the system provides a focused and actionable path for improvement without overwhelming the player.

5 CONCLUSIONS

This project serves as an excellent foundation for developing a more advanced system. However, there are some minor issues, such as the fact that *Edoardo_1-bad-cut_1.csv*, a purposely bad shot, is evaluated as a relatively good shot. This is because a proper evaluation of a good shot involves considering many factors, such as the spin of the ball generated by the wrists and fingers, the trajectory of the ball, the power applied to the shot, and other subtle details.
It is clear that every player has their own unique shooting style. For this reason, this project is designed to assist amateur players who have not yet developed their personal shooting style. At this early stage, when technique and fundamentals are crucial, and there has been no time to adapt the shot to their own physical characteristics, this system can provide valuable guidance.

5.1 Future improvements

Our system is currently very simple and can be enhanced in several ways, given sufficient time and resources. We have outlined a list of potential improvements:

- **Machine Learning**: by collecting the outcome of each free throw (made or missed), it would be possible to integrate a machine learning system that learns which movements are more efficient for scoring. This would result in a more personalized system, eliminating the need for a gold standard comparison. Additionally, such a system could automatically detect the start and end moments of the shot, removing the need for manual cutting of the videos.

- **Ball tracking**: including the trajectory of the ball as a metric would add valuable information to the analysis of the overall movement. For example, differences between various shot trajectories could be computed. Although we attempted to implement this feature, the Qualisys system was unable to simultaneously track both the ball and the body.

- **Improved and additional metrics**: the current system does not account for left-handed players or variations in movement when jumping (since the pelvises are always perfectly aligned). With additional time and effort, these features could be incorporated to make the system more robust and inclusive.

References

[1] *Project Repository.* URL: https://github.com/edofiore/TraBasT-SportTech.

[2] *Motion Capture.* URL: https://en.wikipedia.org/wiki/Motion_capture.

[3] *Optical Marker-Based Motion Capture.* URL: https://www.motionanalysis.com/biomechanics/what-is-optical-motion-capture.

[4] *Qualisys*. URL: https://www.qualisys.com.

[5] *MatPlotLib Library*. URL: https://matplotlib.org/.

[6] *Open3D Library*. URL: https://www.open3d.org/.

[7] *Squat Analysis*. URL: https://github.com/paolo-aliprandi/Squat-Analysis-CVproj.

[8] Rudolph Emil Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[9] *Motion Capture Computer Vision Project*. URL: https://github.com/andy295/Computer-Vision-Project.