Basketball Shooting Analysis via 3D Pose Estimation

Yanjun Chen
Department of Computer Science
Stanford University
yanjunc@stanford.edu

Hanson Lu
Department of Computer Science
Stanford University
hansonlu@stanford.edu

Abstract

There have been successful applications of 3D pose estimation in sports and fitness, for the purpose of technical analysis as well as user-oriented products for real-time training. We would like to explore potential applications of 3D pose estimation in analyzing basketball shooting videos. We envision the application to take two videos of basketball shooting, one as the student video, and one as the teacher video. First we apply an existing 3D pose estimation model, VideoPose3D, to get an pose represented by a skeleton in 3D space. Next, we address the main challenge in this project, which is to align frames in the student video with the teacher video so that they can be well compared and analyzed. For this we apply the Dynamic Time Warping algorithm, and experiment with different distance measures and transformation techniques to improve the alignment. Our final result is a visualization tool that provides frame-by-frame motion analysis and recommendations to the user to match the shooting pose of the teacher.

1. Introduction

The rapid growth of machine learning and deep learning in recent years has greatly pushed the frontiers of many computer vision tasks. Human pose estimation is one of the challenging tasks that gained tremendous progress with the help of deep neural networks. The key idea of human pose estimation is to understand and identify people's poses and movements from raw videos and images. By defining keypoints (joints) on a human body including wrists, elbows, knees and ankles, state-of-the-art pose estimation models could efficiently find the 2D or 3D coordinates of these keypoints in the camera space, effectively extracting a skeleton that captures all the movements.

One of the most popular applications of pose estimation is in AI fitness and sports. In real-world fitness/sports coaching sessions, a human teacher would teach a training exercise and correct students' movements. For example, a fitness instructor may correct your arm angle during pushups, or monitor your knee position in squatting. With pose estimation, however, an AI teacher could automatically detect a student's poses during exercises and identify if any action deviates from the standardized one. This enables students to still learn the correct training movements without the need of a human instructor or wearable fitness equipment.

Many of the existing work and projects that use 3D pose estimation in fitness and sports have focused on physical training. In this project, we want to apply pose estimation to the field of basketball shooting analysis. Shooting in basketball involves a series of body movements including chest, arms, elbow and wrists. Research in basketball has shown that certain shooting pose is scientifically more stable and easier to make shots than the others. A famous shooting form is "three 90 degrees", which refers to the pose where the angles between wrist and forarm , forearm and upper arm, upper arm and torso are all 90 degrees in the shooting preparation mode. As basketball amateur, however, it is really hard to find these angles between body parts, or to diagnose problems of our shooting poses without the help of an instructor. What if there's a tool that could help players learn the correct shooting pose of a professional basketball player with only his/her shooting footage? The great usability of such possibility become the main motivation for this project.

1.1. Problem Statement

In this project, we want to build a system for automatic basketball shooting analysis that helps players learn correct basketball shooting poses on their own. The system will takes as input one video of a "student" shooting basketballs, and another video of a "teacher" demonstrating the correct shooting poses. Then, it would give out quantitative analysis and recommendations to the user for how to improve. The full pipeline will consist of the following 3 key components.

1. Given input videos from the student and the teacher,

estimate and reconstruct their 3D poses. Calculate measurements that are essential to basketball shooting like angles between forearm and upper arm, angles between upper arm and torso, etc.

2. Align shooting poses between student and teacher in an unsupervised manner. After alignment the system should be able to match the shooting process in both videos.

3. Compare the 3D poses of the student with that of the teacher frame by frame. Make concrete recommendations on how to imitate the teacher better.

2. Prior Work

Basketball AI analysis. We have found this repository on github[1] that performs a similar task as ours, but it only uses 2D pose estimation methods and lack the teacher-student comparison feature. This serves as one of our initial sources of inspiration, but we plan to implement our own methods with 3D pose estimation to analyze basketball shooting.

3D pose detection from video. There already exist successful models for single-person 3D pose detection from a single video view. One is VideoPose3D [6]. It is based on convolutional neural networks that take in video as input, and predicts the 3D coordinates of points in a skeleton model. VIBE [4] is another such model, but it uses a GAN-based method, with a discriminator that encourages its proposed poses be similar to in-the-wild 3D motion data. VIBE uses the SMPL body model [5] which is much more complex and nuanced compared to skeletal models. ExPose [2] aims to incorporate accurate facial and hand feature detection together with body 3D pose estimation in the SMPL body model format. However they only use single images instead of temporal data.

3D pose similarity metrics. We have found several metrics for pose similarity during our literature review. First, we may consider evaluation metrics utilized in 3D pose estimation tasks, one of which is *mean per joint position error* (MPJPE) [7] and *Mean Per Joint Angle Error.* [1] develop a new metric based on rich features derived from geometrical properties of joints in the skeleton model, and introduced a semi-supervised learned component where the metric maximizes the similarity score between poses that were labeled similar by human annotators.

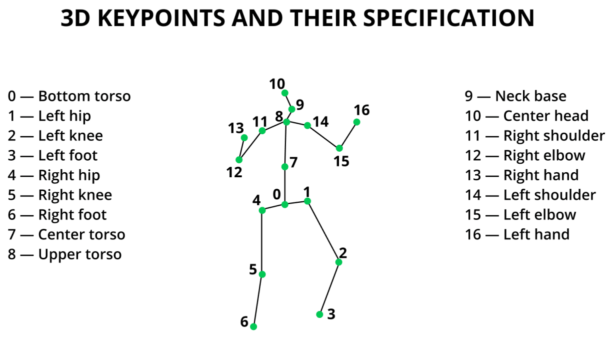


Figure 1: 17-joint h36m skeleton format

Application of 3D pose detection in AI fitness apps. Private companies have designed fitness training apps that incorporate 3D pose detection models to monitor and correct user's form and technique during training. One example is infiGro[2], which performs 3D pose detection using real-time video from a phone camera and provides feedback to the user. This app focuses on physical training in general, while we focus on analyzing basketball shooting, which requires additional localization of the ball and objects in the court such as the hoop.

3. Approach

3.1. 3D Pose Estimation from Video

Data Source Due to the limit availability of dataset with labeled pose skeleton and the difficulty of manual labeling, we are unable to find or collect any labeled dataset for basketball shooting. Even worse, public unlabelled dataset for basketball shooting doesn't exist as well. Processing and collecting one from raw basketball game clips would be a huge engineering work and thus is out of scope for this project. As a result, we will not do any learning on the 3D pose estimation model. Instead, we focus on collecting unlabeled data for qualitative evaluation.

Model Selection As discussed in the prior work section, there exists successful models for single-person 3D pose detection from a video. After careful consideration, we have selected a pre-trained model from the VideoPose3D[6] algorithm published by Facebook research. The algorithm takes in bounding boxes and 2D keypoints predictions from detectron2[8] as input, and uses a CNN architecture with a receptive field of 243 frames. It outputs poses in the 17-joint h36m skeleton format, as shown in Figure 1.

The pre-trained model was trained on the Human3.6M dataset[3], which is a large-scale dataset focusing on 3D human poses. Human3.6M contains 17 different human actions, including walking, eating, smoking and sitting. Unfortunately, none of these activities are close enough to bas-

[1] `https://github.com/chonyy/AI-basketball-analysis`

[2] `https://www.infivolve.com/`

ketball shooting, but the model does show ability to generalize to different human activities. The paper reports an average *mean per joint position error* (MPJPE) of 46.8mm across different human activities in the Human3.6M dataset. We have also qualitatively tests out its performance on the basketball shooting clips we collect and confirm its performance. Please refer to the Experiment section for qualitative results.

3.2. Pose re-orientation

The poses predicted by Video3D for the student and teacher model may be facing in different directions, so therefore we would like to re-orient the poses to face in a similar direction for better analysis. We assume that the root position (the position of the lower torso skeleton joint, index 0 in Figure 1) is at (0, 0, 0). We apply a rotational transform $R = R_t R_n$ to both skeletons.

We compute $\bar{n}$, which is the normalized direction of the normal of the plane formed by joints index 1, 4, 7 (left and right hip, center torso), averaged over all time frames. We compute R_n which rotates $\bar{n}$ to $(0, 1, 0)$. After rotating the skeleton by R_n, we compute $\bar{t}$, which is the average direction of the vector from the lower torso to the middle torso over time frames. We compute R_t, which rotates $\bar{t}$ to $(0, 0, 1)$ and apply it to the entire skeleton.

To compute the rotation matrix R that transforms a unit vector a to unit vector b, we use the following formula:

$$R = I + [v]_\times + \frac{1}{1 + a \cdot b}[v]_\times^2$$

where $v = a \times b$ and $[v]_\times$ is the skew-symmetric cross-product matrix of v.

The result of applying transformation R is that the skeleton will be upright and facing in roughly the same direction. The above transformation works

3.3. Angle Computation

Using the predicted joint 3-D coordinates from VideoPose3D, we implemented a script that computes the angle of every three consecutive joints i, j, k in the skeleton model, where j is the index of the center joint:

$$\theta_{ijk} = \arccos\left(\frac{(\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) \cdot (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_j)}{||\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j||||\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_j||}\right) \cdot \frac{180^\circ}{\pi}$$

where $\hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j, \hat{\mathbf{p}}_k$ denote the predicted joint positions of the model for joint indices i, j, k. For instance, to compute the right elbow angle, we would look at joints 11, 12, and 13.

3.4. Student-teacher pose alignment

3.4.1 Problem Statement

After obtaining per-frame 3D pose estimations for both student and teacher, we need to find a way to effectively compare shooting poses of the student versus that of the teacher. The challenge comes as basketball shooting is consisted of a series of different actions: players usually hold the ball at a low position as preparation and move the ball over their head as they release their shots. Given two user-uploaded basketball shooting videos, it is impossible that players in the two videos match their shooting movements frame-by-frame. It would also be extremely user-unfriendly if we require it. In order to compare shooting poses and give recommendations frame-by-frame, we need to align shooting poses between student and teacher, which we called the pose alignment problem.

Pose alignment is a challenging task: the two input videos can have different length and number of frames; different players also shoot at different speed. Unfortunately, there's very few systematic study or research for this problem. Nor does there exist any labelled dataset for it. As a result, we come up with a novel solution and a novel evaluation method.

3.4.2 Dynamic Time Warping

For solution to the pose alignment problem, we use an algorithm in time series analysis called Dynamic Time Warping (DTW). DTW is a method that calculates an optimal match between two given sequences of different length and speed. In our setting, the algorithm works as this:

Given two sequences of poses $\{p_1, \ldots, p_n\}$, $\{q_1, \ldots, q_m\}$, for each frame in the student video $1 \le i \le n$, DTW assigns a frame in the teacher video $1 \le j \le m$ such that j is monotonically increasing for i and distance(p_i, q_j) is minimized.

DTW is implemented using dynamic programming, similar to finding minimum edit distance. We directly use the `dtw` package in Python.

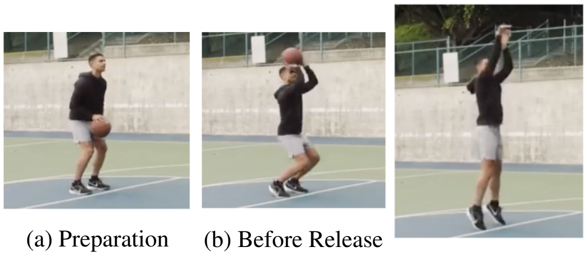
3.4.3 3D pose similarity metrics

The DTW algorithm requires a distance function distance(p_i, q_j) for two poses p_i and q_j. For this, we consider 3 widely used pose similarity metrics from pose estimation literature: cosine distance, *mean per joint position error* (MPJPE) and *mean per joint angle error* (MPJAE).

For each distance metric, p_i and q_j will be a set of hand-selected or derived features from the coordinates of skeleton joints at frame i and j in the two videos respectively. Note that MPJPE and MPJAE are originally used to compare a model's predicted pose with a ground truth pose, but since the metrics are symmetric we use them here as distance measures.

1. Cosine distance

$$d_{cosine} = 1 - \frac{\hat{\mathbf{p}}^s \cdot \hat{\mathbf{p}}^t}{||\hat{\mathbf{p}}^s||||\hat{\mathbf{p}}^t||}$$



(a) Preparation (b) Before Release (c) After Release

Figure 2: 3 phrases of basketball shooting

where $\hat{\mathbf{p}}^s$, $\hat{\mathbf{p}}^t$ denote the predicted joint positions for student and teacher. We used 11 joints in the upper body (head, torso, left and right arms) for distance calculation to better represent basketball shooting movements. This is a naive distance metric and is used as the baseline.

2. MPJPE

$$d_{\text{MPJPE}} = \frac{1}{N}\sum_{j=1}^{N} ||\hat{\mathbf{p}}_j^s - \hat{\mathbf{p}}_j^t||_2$$

where $\hat{\mathbf{p}}_j^s$, $\hat{\mathbf{p}}_j^t$ denote the predicted joint positions of the model for joint index j. We picked the same 11 joints in the upper body as above.

3. MPJAE

$$d_{\text{MPJAE}} = \frac{1}{|S|}\sum_{i,j,k\in S} |\theta_{ijk}^s - \theta_{ijk}^t|$$

where S is a set of 7 important consecutive joint triplets[3] that we believe best represent basketball shooting movements. θ_{ijk} is computed using the formula in section 3.3 angle computation.

3.5. Pose Analysis and Recommendation

After pose alignment, we could now do frame-by-frame pose comparison and analysis. We compute the difference of key joint angles between the student and the teacher as a recommendation to the user. Please refer to the next section for demonstation and qualitative results.

4. Experiments and Demo

We built a full pipeline for basketball shooting analysis and recommendations given a teacher video and a student video. We also build several visualization tool to help visualize the results. In this section, we provide screenshots of our analysis videos as demonstration and qualitative results. We also provide quantitative results for our pose alignment algorithm using our proposed evaluation metric.

4.1. Data Source

Due to the limit availability of dataset for single-player basketball shooting video, we decided to manually find related videos on Youtube, download them and clip them to the appropriate size. We collected a total of 10 video clips of people shooting basketball from different angles. 5 of the 10 videos are from amateur players while the other 5 of them are clips from famous professional basketball shooters like Stephen Curry, Steve Nash and Ray Allen.

4.2. 3D Pose Estimation and Angle Computation

We show the results of our visualization in figures 3, which is a screenshot of a video that presents the predicted 3d pose reconstruction alongside the original video clip.

From these results we see that the value of the angle calculation is generally consistent with the movement of the basketball player – the angle of the right elbow increases as the player prepares for and then performs the shot.

4.3. Evaluating Student-teacher Pose Alignment

4.3.1 Establishing a metric

Evaluation of pose alignment is hard: there doesn't exist publicly available labelled dataset to use. Also, hand labeling pose matches frame by frame is extremely tedious and time-consuming. To mitigate this problem, we proposed a simple metric that evaluates how well a method matches key frames of two basketball shooting videos. Key frames are defined using on our domain knowledge as below.

Each basketball shooting movement can be approximately dissected into 3 phrases: preparation, before release, and after release. An example of these 3 phrases is shown in Figure 2. We define the preparation frame to be the last frame before the player tries to move the ball upward; we define the before release frame to be the last frame before the player tries to move the forearm and push the ball out; we define the after release frame to be first frame that the ball leaves the hand while the arm fully extends. This setup helps us capture the entire shooting process by using only 3 key poses. Frames between these key frames are a series of transitional movements that follow naturally. If a method is able to match the 3 key frames perfectly, it's very likely that all the transitional poses in between are also matched nicely. We obtain these key frames by manual annotation.

Given 3 key frames for each video, we now define *Mean Distance to Key Frames* (MDKF), the evaluation metric for pose alignment as below:

$$\text{MDKF} = \frac{1}{K}\sum_{k=1}^{K} |f_k^s - \tilde{f}_k^t| + |f_k^t - \tilde{f}_k^s|$$

[3] 1) neck, chest, and right shoulder; 2) neck, chest, right shoulder; 3) left, chest, and right shoulder; 4) chest, left shoulder, and elbow; 5) chest, right shoulder, and elbow; 6) left elbow; 7) right elbow

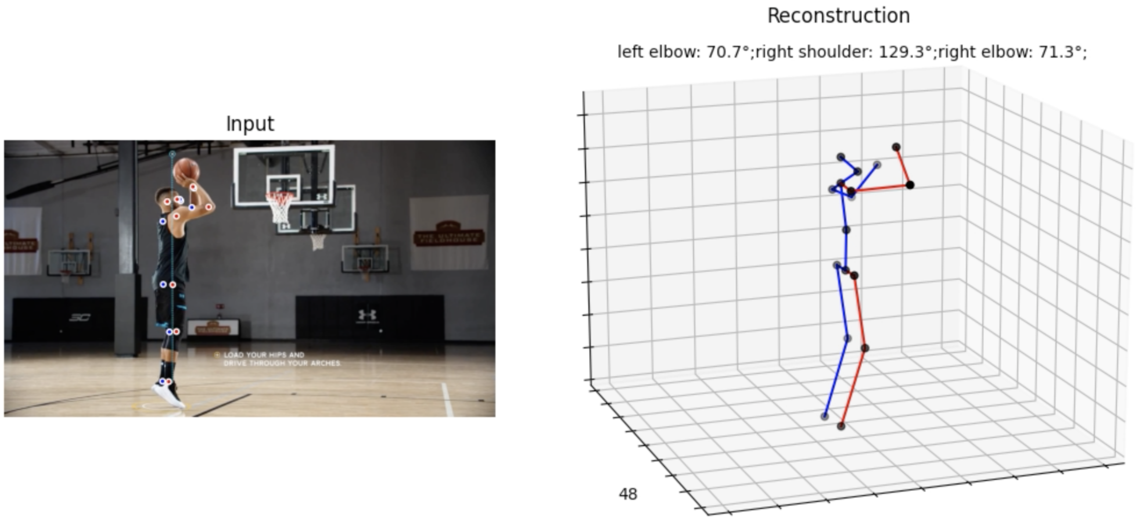


Figure 3: Qualitative results for 3D pose estimation and angle computation (without pose re-orientation)

where $K = 3$ is the number of key frames used; f_k^s, f_k^t are the ground-truth frame indices of key frame k for student and teacher; and $\tilde{f}_k^s$, $\tilde{f}_k^t$ are the matched frame indices of key frame k from student and teacher. The smaller MDKF is, the more accurate the corresponding pose alignment is.

4.3.2 Results for alignment

	Cosine Distance	MPJPE	MPJAE
w/o re-orient.	11.67	14.24	9.85
w/ re-orient.	10.95	10.65	9.85

Table 1: Mean Distance to Key Frame (MDKF) for 3 distance metrics used in the DTW algorithm

To evaluate our pose alignment algorithm, we annotated the 3 key frames for each of the 10 video clips we collected. Since our metric is defined on video pairs, these 10 videos give us $10*9/2 = 45$ video pairs for evaluation. We run Dynamic Time Wrapping with 3 different pose distance metrics introduced above and calculate the average MDKF. The results are shown in Table 1.

As shown in the table, *Mean Per Joint Angle Error* (MPJAE) performs the best. We also see that our pose-reorientation technique greatly improved the alignment results of position-based distance functions.

4.4. Qualitative results

In Figure 4 we present a screen-shot of the final video after performing student-teacher pose alignment and re-orienting the poses.

In this figure, skeletal components on the right-hand side are colored red, while ones on the left-hand side are colored blue. The points on the original video indicate predictions of the 2D joint prediction component of VideoPose3D, and are not necessarily projections of the joints in the skeleton onto the image frame, since the 3D joint prediction component may account for noise in the former component.

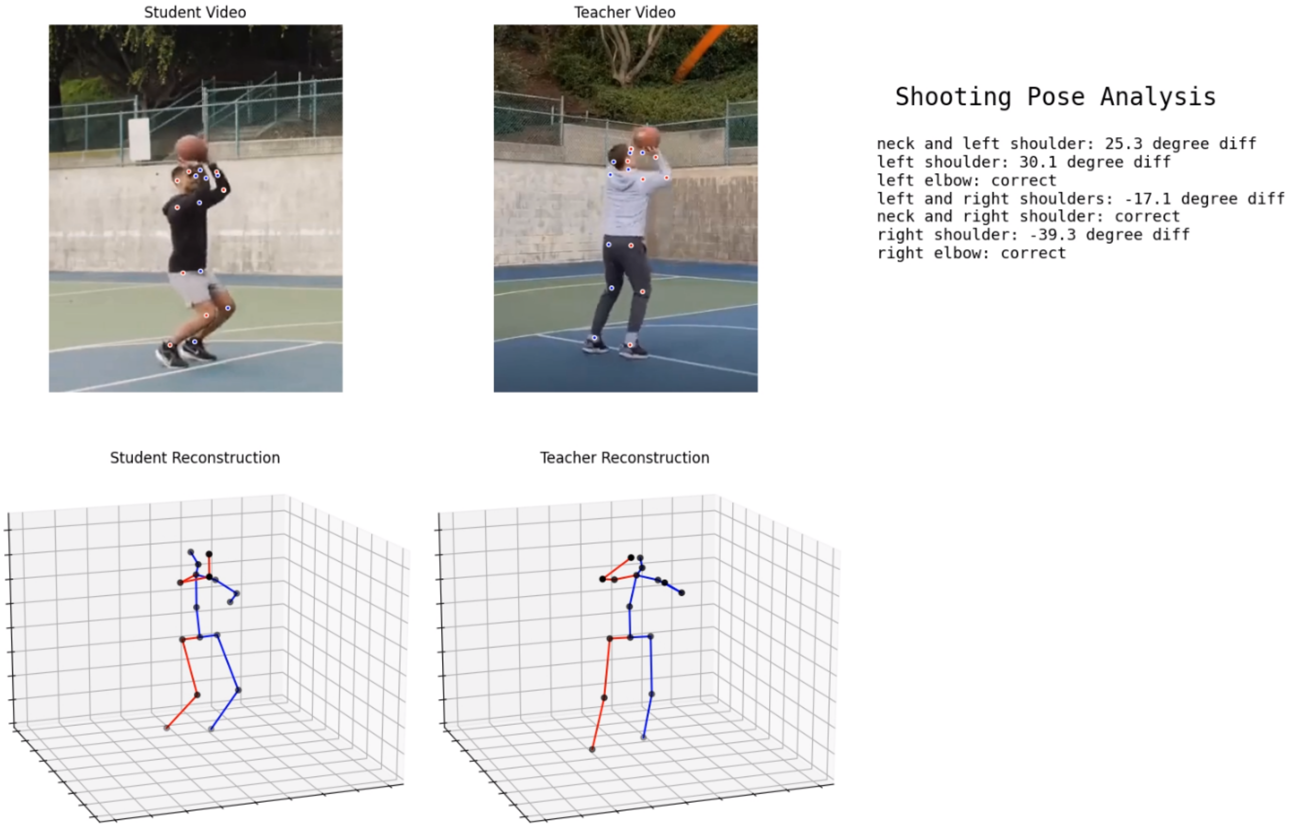
Throughout the video, a user may pause at a specific frame to compare their pose with a reference pose. As seen in the figure, the teacher is facing a different direction than the student, but our re-orientation method mitigates this issue by presenting both poses in the same direction, making a visual comparison of the poses much easier. We also see that the poses are aligned properly, given the fact that the two frames in the screen shot may be at differing time points in their original videos.

In addition, we have computed the angles of certain important joints and show it to the user, which provide better guidance on how to adjust their pose to match that of the teachers, in combination with the visualization of the skeletons.

5. Conclusion

In conclusion, we have successfully applied an existing video 3D pose estimation method to analyze basketball shooting videos, and have constructed an effective method to align estimated poses from a student and teacher video.

Nevertheless, we acknowledge the limited scope of our project, and our method is limited by the effectiveness of



```
Shooting Pose Analysis

neck and left shoulder: 25.3 degree diff
left shoulder: 30.1 degree diff
left elbow: correct
left and right shoulders: -17.1 degree diff
neck and right shoulder: correct
right shoulder: -39.3 degree diff
right elbow: correct
```

Figure 4: Final results after pose re-orientation and alignment.

the 3D pose estimation neural model, which we only use to perform inference. We identify the following possible future directions:

1. The current VideoPose3D skeleton model does not contain hand position and angle, which are important in affecting the trajectory of the basketball; we may need to also include a hand pose estimation pipeline, and include a more close-up video of a player's hand as the input to the pipeline.

2. VideoPose3D only works with a single subject; however other players may appear on basketball courts and it may be difficult to obtain a clip that only contains one player. Some constraints might need to be introduced to limit detection of the most prominent player.

3. Potentially simulate or gather a larger dataset of basketball shooting videos to fine-tune.

4. Explore the possibility of using multiple viewpoints.

5. Add smoothness constraints to the predictions of the pose estimation model.

References

[1] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao. Learning a 3d human pose distance metric from geometric pose descriptor. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1676–1689, 2011.

[2] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black. Monocular expressive body regression through body-driven attention. In *European Conference on Computer Vision*, pages 20–40. Springer, 2020.

[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] M. Kocabas, N. Athanasiou, and M. J. Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020.

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[6] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions

and semi-supervised training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7745–7754, 2019.

[7] W. Z. S. Li and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *International Conference on Computer Vision (ICCV)*, pages 2848–2856, 2015.

[8] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.