Pacific Graphics (2023)
R. Chaine, Z. Deng, and M. H. Kim (Editors)

Reconstructing Baseball Pitching Motions from Video

Jiwon Kim, Dongkwon Kim, and Ri Yu

Ajou University, South Korea

Abstract

Baseball is one of the most loved sports in the world. In baseball game, the pitcher's control ability is a key factor for determining the outcome of the game. There are a lot of video data shooting baseball games, and learning baseball pitching motions from video can be possible thanks to the pose estimation techniques. However, reconstructing pitching motions using pose estimators is challenging. When we watch a baseball game, motion blur occurs inevitably because the pitcher throws a ball into the strike zone as fast as possible. To tackle this problem, We propose a framework using physics simulation and deep reinforcement learning to reconstruct baseball pitching motions based on unsatisfactory poses estimated from video. We set the target point and design rewards to encourage the character to throw the ball to the target point. Consequently, we can reconstruct plausible pitching motion.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation; Reinforcement learning;***

1. Introduction

Baseball is a sport loved by people all over the world. There are many video data shooting the baseball games, and baseball coaches and players analyze opponents' performance or review their playing watching the videos. Meanwhile, researches reconstructing 3D human motions from 2D video have been developed remarkably [PKM*18, YPL19, ZYM*23]. However, the reconstructed motion quality is highly dependent of the accuracy of pose estimation techniques. Although many issues exist for more precise pose estimation from 2D videos, motion blur is one critical issue to reconstruct highly dynamic and fast motions, such as pitching or swing. In baseball game videos, the pitcher swings his arm too quickly, making it difficult to estimate the arm posture well as shown in Figure 1.

Figure 1: *Motion blur occurred when baseball pitching which degrades the performance of the pose estimator (VIBE [KAB20]).*

We propose a framework to reconstruct baseball pitching motions from 2D video even the pose estimation results are unsatisfactory. Our framework takes advantage of using the reinforcement learning and the physics simulation to reproduce realistic motions according to the given goal. The pitcher must throw a ball into the strike zone. We set a target point signifying the center of the strike zone, and design the two reward terms which encourage the character to throw the ball toward the target point as fast as possible.

2. Method

We adopted the framework proposed by Yu et al. [YPL21] and revised it. When an input video is given, it utilizes OpenPose [CHS*19] to obtain 2D joint positions which are used to extract contact information. 3D estimated poses are obtained using VIBE [KAB20]. Finally, using deep reinforcement learning, the motion following the hints earned from the video is reproduced.

2.1. Environment Setup

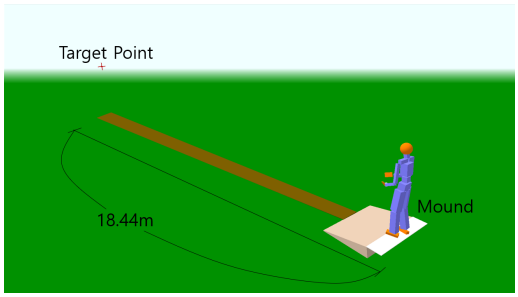

Figure 2: *The pitcher's mound and a target point.*

A baseball pitcher starts pitching standing on the mound, which

has the highest location in the field. We approximate the pitcher's mound as a box tilted about 0.2 radian to form a down-slope toward the home plate. We set a target point reaching a distance of 18.44 meters from the mound to the home plate (See Figure 2). We put a ball as a weld joint of the pitching hand and let the character release the ball when the time comes.

2.2. Rewards

The framework proposed by Yu et al. [YPL21] uses five reward terms r_{base} to make the virtual character imitate the high-level hints, such as poses and contact information extracted from the input video.

$$r_{\text{base}} = w_q r_q + w_v r_v + w_{\text{ori}} r_{\text{ori}} + w_{\text{contact}} r_{\text{contact}} + w_{\text{reg}} r_{\text{reg}} \quad (1)$$

The pose and velocity rewards r_q and r_v encourage to mimic the estimated poses. However, in case of the pitching example, estimated arm poses are not reliable because of motion blur. Hence, we reduced the tracking weight of the arm joints 0.1 times.

To give the ability to control the ball, we added two reward terms, r_{ballPos} and r_{ballDir}. The first term r_{ballPos} is for reducing the distance between ball and the target point and the second term r_{ballDir} is for matching the throwing direction of the ball toward the target point.

Ball Position. When a ball is released, its trajectory is predictable because it moves along the curve. Therefore, we can calculate the expected position of the ball at every frame after the release point. The ball position reward encourages the character to throw the ball to make a ball reaching the target position as closely as possible.

$$r_{\text{ballPos}} = \exp\left(-\alpha_{\text{ballPos}} \|d\|^2\right), \quad (2)$$

where d is the distance from the ball and the target point.

Ball Direction. The pitcher should throw a ball toward the strike zone. We devise the ball direction reward term which minimizes the difference between two vectors $\vec{p}$ and $\vec{q}$, where $\vec{p}$ is the CoM speed of the ball and $\vec{q}$ is a vector from the ball and the target point.

$$r_{\text{ballDir}} = \exp\left(-\alpha_{\text{ballDir}} \|1 - \vec{p} \cdot \vec{q}\|^2\right) \quad (3)$$

We use the reward $r = r_{\text{base}} + w_{\text{ballPos}} r_{\text{ballPos}} + w_{\text{ballDir}} r_{\text{ballDir}}$ for reinforcement learning. Note that r_{ballPos} and r_{ballDir} are zero before releasing the ball.

3. Results

We used the video clip of the major league game as an input video. Figure 3 shows the reconstruction results of the previous framework [YPL21] and ours. Although the input is overhand throwing pitcher's video, the arm motion reconstructed using [YPL21] is not the same and the character cannot throw the ball as in the video (Figure 3 top). It's because the controller is trained to imitate the reference motion based on incorrectly estimated poses due to the motion blur. On the other hand, our framework successfully reproduced overhand pitching motion even though we used the same reference motion (Figure 3 bottom).

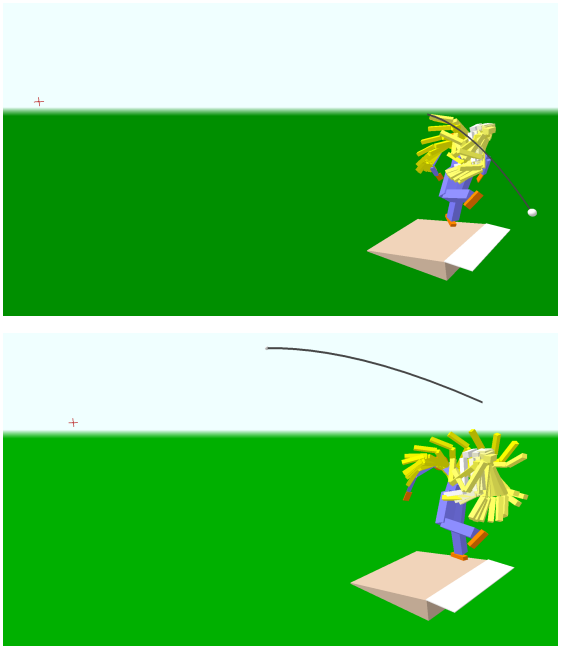

Figure 3: *Reconstructed results of a pitching motion using the previous framework [YPL21] (first row) and using our framework (second row)*

4. Discussion

Our framework shows the potential to reconstruct human motions based on the estimated poses from off-the-shelf pose estimators which suffer from various reasons, such as motion blur and pose ambiguities. Currently, our character doesn't have sufficient control over the speed or position of the ball. In the future work, we will adopt curriculum learning to enhance the control ability of the virtual pitcher. Furthermore, we plan to collect pitching videos and reconstruct pitching motions to built the data set. Based on that, we will train a model to learn a variety of pitching style.

References

[CHS*19] CAO Z., HIDALGO MARTINEZ G., SIMON T., WEI S., SHEIKH Y. A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). 1

[KAB20] KOCABAS M., ATHANASIOU N., BLACK M. J.: Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 5253–5263. 1

[PKM*18] PENG X. B., KANAZAWA A., MALIK J., ABBEEL P., LEVINE S.: Sfv: Reinforcement learning of physical skills from videos. *ACM TOG 37*, 6 (2018), 1–14. 1

[YPL19] YU R., PARK H., LEE J.: Figure skating simulation from video. *Computer Graphics Forum 38*, 7 (2019), 225–234. 1

[YPL21] YU R., PARK H., LEE J.: Human dynamics from monocular video with dynamic camera movements. *ACM TOG 40*, 6 (2021). 1, 2

[ZYM*23] ZHANG H., YUAN Y., MAKOVIYCHUK V., GUO Y., FIDLER S., PENG X. B., FATAHALIAN K.: Learning physically simulated tennis skills from broadcast videos. *ACM TOG 42*, 4 (2023), 1–14. 1