MV-Sports: A Motion and Vision Sensor Integration-Based Sports Analysis System

Cheng Zhang[†*], Fan Yang[‡], Gang Li[†*], Qiang Zhai[‡], Yi Jiang[‡], and Dong Xuan[†]
[†]Department of Computer Science and Engineering, The Ohio State University
[‡]DeepCode Robotics Co., Ltd.
{zhang.7804, li.2384, xuan.3}@osu.edu, {fanyang, qiangzhai, yijiang}@deepcode.cc

***Abstract*—Recently, intelligent sports analytics is becoming a hot area in both industry and academia for coaching, practicing tactic and technical analysis. With the growing trend of bringing sports analytics to live broadcasting, sports robots and common playfield, a low cost system that is easy to deploy and performs real-time and accurate sports analytics is very desirable. However, existing systems, such as Hawk-Eye, cannot satisfy these requirements due to various factors. In this paper, we present MV-Sports, a cost-effective system for real-time sports analysis based on motion and vision sensor integration. Taking tennis as a case study, we aim to recognize player shot types and measure ball states. For fine-grained player action recognition, we leverage motion signal for fast action highlighting and propose a long short term memory (LSTM)-based framework to integrate MV data for training and classification. For ball state measurement, we compute the initial ball state via motion sensing and devise an extended kalman filter (EKF)-based approach to combine ball motion physics-based tracking and vision positioning-based tracking to get more accurate ball state. We implement MV-Sports on commercial off-the-shelf (COTS) devices and conduct real-world experiments to evaluate the performance of our system. The results show our approach can achieve accurate player action recognition and ball state measurement with sub-second latency.**

I. Introduction

Nowadays, the rapid advancement of mobile networks, wearables, and artificial intelligence technologies is bringing digital innovations to the sports arena. Many elite sports clubs are using sports analytics to gain understanding of their players' performance for coaching, tactic, and technical analysis. For example, the ProZone system has been installed at Old Trafford in Manchester and Reebook Stadium in Bolton for players' motion analysis [1]. Besides professional clubs, live sports broadcasting is adopting sports analytics to present online game statistics to viewers [2]. With the coming robots era, sports robots [3] developed to rival humans would also benefit from sports analytics for prediction, decision making, and motion planning. In these applications, real-time analysis is indispensable. In addition, there is also a growing trend of using sports analytics among amateurs on common playfield, where an affordable and easy-to-deploy system is preferable. Therefore, a low-cost system that performs real-time and accurate sports analysis is very desirable and has many important applications in real-world sports activities.

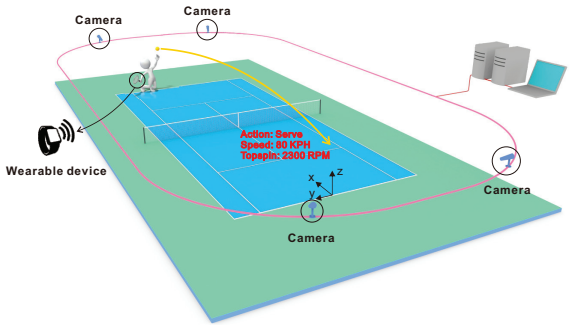

Fig. 1. MV-Sports' working scenario. See text for more details.

A majority of existing systems for sports analytics are built with cameras [4], [5]. Hawk-Eye [6] is the most representative technology that employs 6 to 10 high performance cameras to equip the arena for 3D object tracking. The tracking results are obtained seconds after the game and mainly used in replays for commentators. Although Hawk-Eye has good accuracy, it is too expensive to afford for common playfield. Besides, the tracking is not real-time and thus is unsuitable for applications that regard online analysis. Some systems embed wireless sensors inside objects such as balls [7] to provide real-time analytics. However, this is too intrusive for broad adoption.

In this paper, we propose MV-Sports, a cost-effective system built on commercial off-the-shelf (COTS) wearable **M**otion sensors and **V**ision sensors (i.e., cameras) for sports analysis. We use tennis as a case study and aim to analyze movements of both the player and the ball. Fig. 1 shows MV-Sports' working scenario in a tennis court. The setup is easy and non-intrusive as the player only needs to don a wearable device and place two cameras near each base line instead of modifying the ball or racket. The rationale for using both motion and vision sensors is their complementary characteristics [8], [9]. The motion data (M) data is more lightweight and the visual data (V) is more precise. By integrating M and V data, MV-Sports has great potential to achieve real-time and accurate analysis.

In our case study, we first aim to analyze players' shots, which entails recognizing players' fine-grained actions such as serves, forehands, backhands, etc. The motion-based approach is fast, but not robust due to the noise caused by successive

[*] This work was performed while Cheng Zhang and Gang Li were interns at DeepCode Robotics collaborating with the other authors.

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

movements. Visual data can provide more accurate analysis thanks to their rich content. However, visual processing is more time-consuming, especially for differentiating fine-grained actions. To achieve fast and accurate action recognition, we first leverage motion signals to quickly highlight the meaningful content for action detection, avoiding redundant processing. Then we integrate the highlighted M and V data via a uniform neural network to train an end-to-end model for classification. To capture spatiotemporal information of the action, we use a long short term memory (LSTM) network to learn M and V features.

The second task is to continuously measure ball states, including ball position, speed, and spin. Two preliminary approaches can be used for ball state tracking. The ball motion physics-based method is robust against environmental noise, but requires accurate initial state, which is difficult to measure. The vision positioning-based technology is able to quickly rectify the noisy initial state, but sensitive to environmental factors especially when using COTS cameras under practical constraints. To closely integrate the results from the two methods for more accurate tracking, we formulate the ball state tracking as an extended kalman filter (EKF) problem with information extracted from both tracking methods. Specifically, we measure the initial ball state based on M data and use it to initialize the ball motion physics-based tracking. Meanwhile, we use the observed ball positions from V data to update EKF for more accurate ball state measurement at each point in time.

We prototype MV-Sports in a tennis court on commodity wearable devices and RGB cameras with one laptop as the back-end. We conduct extensive experiments to recognize five typical actions in tennis: forehand topspin, forehand slice, backhand topspin, backhand slice, and serve. Results from 2,500 video clips and corresponding motion samples achieve average classification accuracy of 98%, which outperforms conventional pure motion-based or pure vision-based methods. For ball state measurement, we evaluate MV-Sports over 100 different shots with 40,000 video frames. The results show that our system achieves median errors of 0.5 m/s for speed measurement and median errors of 40 revolutions per minute (RPM) for spin measurement. Both player action recognition and ball state measurement are conducted with sub-second latency, which is real-time. All the results demonstrate the effectiveness and efficiency of our system.

In this study, we claim the following contributions:

- We design MV-Sports, a low cost system that is easy to deploy and provides real-time and accurate sports analysis.
- We propose a deep learning-based framework to integrate motion and visual data for fast and accurate player action recognition. Besides, we devise an EKF-based approach to formulate ball state measurement and combine motion and visual data for precise ball state tracking.
- We implement MV-Sports in a tennis court with COTS devices. Extensive real-world experiments are conducted to evaluate MV-Sports' performance and the results demonstrate the accuracy and efficiency of our system.

In summary, we have developed a cost-effective sports analysis system by integrating motion and vision sensors. Although in this work, we focus on two tasks in tennis, player action recognition and ball state measurement, our methodology can be used to obtain more information such as player action consistency and intensity. Besides tennis, our MV-Sports can be easily extended to other sports with modest modifications, such as golf, baseball, volleyball, badminton, and table tennis.

The remainder of the paper is organized as follows. Section II reviews related work. Section III elaborates the design of MV-Sports system. Section IV presents the system implementation and reports our experimental evaluation. Finally, Section V concludes the paper.

II. Related Work

In this section, we describe three categories of related research work.

Sports Analysis Systems: Many sports analysis systems have been built in both industry and academia. For example, Hawk-Eye [6] uses high-speed cameras installed around the field to track ball movements. Bagadus [4] integrates a video capturing system with a sports tracking system for soccer game analytics. In [10], the authors focus on ball detection and tracking algorithms for soccer video analysis. Chen et al [11]. leverage physical characteristics to assist ball trajectory reconstruction for basketball and volleyball video analysis. Babolat [12] and Sony [13] present the tennis player an overview of the game by analyzing data collected from sensors in the racket handle. iBall [7] embeds inertial sensors and ultrawide band radios inside balls and players' shoes for ball tracking and spin analysis in cricket. Zepp [14], a startup in this area, has developed several mobile applications for baseball, softball, golf, and tennis game analysis based on sports sensor data. Unlike these systems, we aim to build a low-cost system that is easy to deploy and performs real-time and accurate sports analysis.

Deep Learning-based Action Recognition: There is a large body of work on vision-based human activity analytics [15]–[18]. Karpathy et al. [15] extend convolutional neural networks (CNNs) for large scale sports video classification. Work in [16] uses deep 3-dimensional convolutional networks for spatiotemporal feature learning in video analysis. The authors in [17], [18] utilize the two-stream CNN architecture that incorporates spatial and temporal networks for action and gait recognition in video data. There are also works on motion-based action recognition. Guan et al. [19] employ deep recurrent LSTM networks for action recognition using wearable sensing data. To enable on-node real-time activity classification, Ravi et al. [20] propose a deep learning approach that combines features learned from inertial sensor data with engineered features. DeepConvLSTM [21] presents a deep learning framework for activity recognition by combining convolutional and LSTM recurrent layers.

Continuous Ball State Tracking: Various approaches have been proposed in the literature for tracking the small, fast-

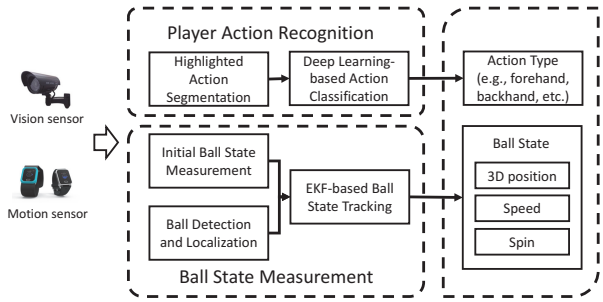

Fig. 2. System workflow of MV-Sports.

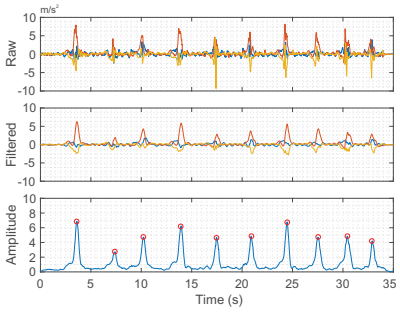

Fig. 3. The raw 3-axis accelerometer M-Stream samples, the smoothed stream with mean filter, and the amplitude of highlighted action segmentation results.

moving ball. Yan et al. have conducted a notable work in this area. They leverage enhanced low-level features and a modified particle filter for tennis ball detection and tracking in low quality sports videos [22] and employ a layered data association method that exploits graph theory for ball tracking in cluttered environments. Work in [23] scores and filters all possible ball trajectories based on their lengths and relationships in a timeline model. Authors in [24] explore trajectory reconstruction by approximating ball travel on tilted planes for ball position measurement. Recently, machine learning has also been exploited for automatic ball tracking such as random forest and neural networks.

III. MV-Sports Design

In this section, we first illustrate the workflow of our MV-Sports system. Next, we discuss the main components in our system.

A. System Overview

As shown in Fig. 2, our system consists of two main components: player action recognition and ball state measurement.

– *Player Action Recognition:* The purpose of this component is to identify the player's shot types such as serve, forehand, backhand, etc. We first process the motion sensor readings to identify the meaningful samples that represent the performed action. Next, we highlight and segment the corresponding video frames in the same time period for further analysis. This method of action highlighting in video data is fast with the help of the lightweight motion data. For action recognition, the meaningful motion data samples and video frames are together fed to a neural network for feature extraction, training, and classification.

– *Ball State Measurement:* Besides analyzing the player's actions, MV-Sports also measures the ball's state including ball position, speed, and spin at each point in time. We first measure the ball speed and spin when the ball is hit by the racket using the motion sensor data. Next, we use the initial state to initialize the ball motion physics-based tracking. Meanwhile, we also perform ball detection and localization on continuous video frames for vision-based tracking. Both tracking results are then fed to the EKF for ball state measurement with higher accuracy.

B. Player Action Recognition

To recognize player's actions, we first leverage motion data to highlight meaningful actions. Then we combine both the motion and the video data for accurate action recognition.

1) Highlighted Action Segmentation: The very first step for player action recognition is to highlight actions. Currently, most camera-based systems mainly rely on high-level event detection methods or low-level features such as optical flow for video highlighting. However, these approaches are not only time-consuming for real-time sports analytics but also too coarse to capture the most meaningful content. In MV-Sports, we explore the motion sensor-based action detection to highlight actions with an adaptive thresholding strategy in motion data streams. Our solution includes two stages. The first stage is highlight sensing. Specifically, we use a sliding window of size W to compute the accumulated amplitude of 3D accelerometer signals:

$$A(t) = \sum_{t}^{t+W} \sqrt{a(t)_x^2 + a(t)_y^2 + a(t)_z^2} \quad (1)$$

where $a(t)_x$, $a(t)_y$, and $a(t)_z$ are recorded 3-dimensional acceleration streams, named M-Stream, at time t. We set a threshold A_{th}. When $A(t)$ exceeds A_{th}, the samples within the window are highlighted to represent an action. Based on our real-world experiments, we set A_{th} as $10\times$ the average amplitude in the window of size W. In our implementation, the sampling frequency of motion sensors is 100 Hz; hence we set the sliding window size W to be 150 samples, which is long enough to capture a complete action. The second stage is action segmentation, where we first calculate the second order derivative of the above highlighted M-Stream. We call the result as dM-Stream. The dM-Stream point whose value equals zero corresponds to the peak point in the real-world action. With the obtained peak point, we segment M-Stream using the peak point as the center with W samples, resulting in the start and end points of the action. Algorithm 1 presents the pseudocode for the overall solution. Fig. 3 shows M-Stream samples and segmented results. As video data are synchronized with the motion data, actions in video data can be quickly highlighted using the timestamps (i.e., point in time) of the start and end points. According to our experiments on 2,500

Algorithm 1 Action Highlight with Motion Stream

1: Compute the filtered motion stream
2: Compute the accumulated amplitude $A(t)$ of M-Stream at time t based on Eq. (1)
3: **if** $A(t)$ is larger than A_{th} in the window of size W **then**
4: Compute the dM-Stream
5: Find zero value in the dM-Stream as the peak point
6: Segment signals with window W to get start point and end point of highlighted action
7: **end if**
8: **return** Start point and end point

action clips, MV-Sports achieves 99.6% action segmentation accuracy.

2) Conventional Action Recognition: For action recognition, there are two conventional approaches: motion-based and vision-based. The motion-based approach entails extracting hand-crafted features from 3D accelerometer and 3D gyroscope data. More specifically, we extract motion features separately from each axis, and then concatenate them as a feature vector for further classification. Although the motion-based method proves to be a promising solution for human action recognition in the literature, it cannot be applied directly in accurate sports analytics for the following reasons. First, motion information can only be treated as low-dimensional temporal data, which are not robust on wide varieties of sports action. Second, successive movements in the action add much noise into raw motion data. Third, the trained model does not work well for players on whom it has not been trained as motion patterns vary among different people. For vision-based action recognition, existing approaches applied to sports are limited. They suffer from performance degradation when differentiating fine-grained actions. This is especially true for sports such as tennis where the variations observed among different shots (backhand and forehand) or sub-types of a shot (forehand topspin and forehand slice) are subtle, making automatic action classification extremely difficult. In order to solve these problems, we propose a deep learning-based MV integration framework for action recognition.

3) Deep Learning-based MV Integration Framework: We choose LSTM, a variant of recurrent neural network (RNN), to integrate MV data for action recognition. LSTM is well known for its superiority in capturing long-term temporal information in various tasks such as natural language processing and video action recognition. Besides, the types of data precessed by LSTM can be one-dimensional signals, text, images, or videos, which provides more opportunities to bridge the "domain gap" between motion and vision data. Thus, the LSTM network is suitable for MV data-based action recognition in the context of this work.

Fig. 4 gives an overview of our proposed LSTM-based MV integration framework for action recognition. First, we locate the moving player in the given video frame at each time point with aggregate channel features (ACF) [25]. Then we utilize the deep convolutional neural network (CNN) to extract spatial visual representations. After that, we feed the extracted visual features and the corresponding hand-crafted M-Stream features at each time point into the LSTM network for further training. Next, LSTM learns combined motion and vision features to capture long-term temporal information of player's behavior in the real-world environment. Finally, the softmax loss layer is used for action pattern classification. In the following, we describe each part in details.

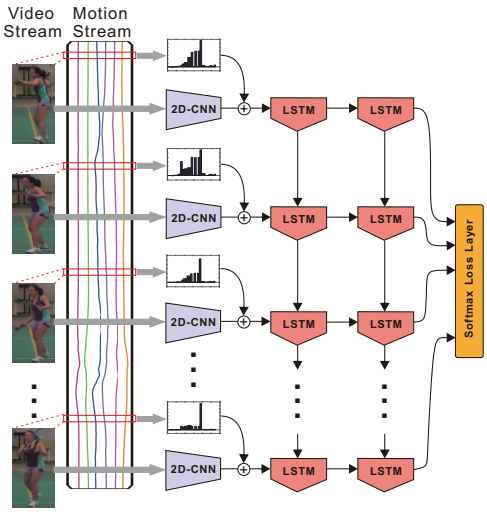

Fig. 4. The LSTM-based MV integration framework for player action recognition.

Spatiotemporal Feature Extraction: The vision-based feature generation includes two parts: player localization and feature extraction. In our framework, we first employ the ACF detector [25] to locate the player within a bounding box in each frame because of its good balance between efficiency and performance. Note that if multiple people exist in the camera view, a motion and vision integration-based method [26] can be used to accurately identify the target player with motion wearables. Secondly, we exploit CNN as the feature extractor to obtain visual clues in the bounding box. Specifically, we take Caffe-BLVC model [27] to initialize the network and employ the network architecture discussed in [28] to extract expressive spatial visual information. For the entire time period of one action with N frames, we fetch the activation in the fully-connected layer 7 in AlexNet [28] for each frame as the final visual features. For motion signals, we compute the 27-dimensional statistical feature vector as described in [20] for the 6-axis M-stream.

Spatiotemporal Information Integration and Training: We first briefly introduce LSTM for those unfamiliar with this technique. The LSTM networks were initially proposed to tackle the vanishing and exploding gradients problem. LSTM networks have special cells that contain four main components: an input gate, a neuron with a self-recurrent connection, a forget gate, and an output gate [29]. The input gate updates the state of the memory cell or blocks it according to the input data. The self-recurrent connection ensures that the state of a memory cell has feedback with a delay of 1 time step.

The forget gate allows the memory cell to remember or forget its previous state by adjusting its self-recurrent connection. Finally, the output gate can allow the state of the memory cell to have an effect on other neurons or prevent it. In summary, they enable the LSTM unit to learn extremely complex and long-term temporal dynamics.

Mathematically, at time t, $\mathbf{x}^t$ and $\mathbf{h}^t$ are the input and output vectors, respectively. In our case, $\mathbf{x}^t$ is the combination of motion and visual feature vectors and $\mathbf{h}^t$ contains the integrated MV information. $\mathbf{W}$ is the input weighted matrix, $\mathbf{R}$ is the recurrent weighted matrix, and $\mathbf{b}$ is the bias vector. The $\sigma(\cdot)$ and $\mathbf{tanh}(\cdot)$ are nonlinear activation functions, which map real values to $(0, 1)$ and $(-1, 1)$. The $\odot$ and $\oplus$ denote the dot product and the sum of two vectors. When given $\mathbf{x}^t$, $\mathbf{h}^{t-1}$ and $\mathbf{c}^{t-1}$, the updating functions in the LSTM cell are as follows:

$$\begin{aligned}
\mathbf{g}^t &= \mathbf{tanh}(\mathbf{w}_g\mathbf{x}^t + \mathbf{R}_g\mathbf{h}^{t-1} + \mathbf{b}_g) \\
\mathbf{i}^t &= \sigma(\mathbf{w}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{h}^{t-1} + \mathbf{b}_i) \\
\mathbf{f}^t &= \sigma(\mathbf{w}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{h}^{t-1} + \mathbf{b}_f) \\
\mathbf{c}^t &= \mathbf{g}^t \odot \mathbf{i}^t + \mathbf{c}^{t-1} \odot \mathbf{f}^t \\
\mathbf{o}^t &= \sigma(\mathbf{w}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{h}^{t-1} + \mathbf{b}_o) \\
\mathbf{h}^t &= \mathbf{tanh}(\mathbf{c}^t) \odot \mathbf{o}^t.
\end{aligned} \tag{2}$$

For action recognition, we aim to learn a nonlinear function to separate the feature space into several subspaces according to the number of categories. Therefore, we apply the softmax loss layer to connect the LSTM cells in the network. Our training algorithm adopts mini-batch stochastic gradient descent for optimizing the objective function. The training data are divided into mini-batches. Training errors are calculated upon each mini-batch in the softmax loss layer and backward propagated to the lower layers; network weights are updated simultaneously. We use the "step" learning rate policy. We initialize the learning rate to 10^{-4} and Gamma to 10^{-1}. Momentum and weight decay are set to 9×10^{-1} and 5×10^{-4}, respectively.

Online Action Classification: In the online testing stage, the motion data are processed for generating motion features and each video frame is sent into a CNN for visual feature extraction. Next, the M features and V features together are fed into the LSTM to compute the feed-forward network based on matrix multiplication for classification. The whole scheme is efficient, which is demonstrated in Section IV.

C. Ball State Measurement

Now we present our approach for ball state measurement. Here, ball state includes ball position, speed, and spin and it changes when the ball is flying or bouncing as the ball motion is affected by several forces such as gravity and air drag. Thus, the ball state is variable and we need to continuously track it during the ball's flight.

Two methods exist for tracking ball state. The first one is based on ball motion physics. Ball state during the flight is calculable, if given the initial state when the ball is hit by a racket. As the ball flies, we can analyze the physical process and calculate ball state at each timestamp. The initial state can be determined by the motion of the racket, which is measured by motion sensors. The accuracy in this method is heavily affected by the accuracy of the initial state as all following states are calculated from it. The initial state, however, cannot be accurately measured due to noise from motion sensors. The other approach is purely vision-based. Visual positioning technologies can continuously track the ball's 3-dimensional location during its flight. As cameras can visually capture the ball very rapidly (60 frames per second in this work), the continuous positions of the ball entail its states over time. However, visual positioning is very sensitive to environmental factors such as lighting conditions. This method is not reliable or robust in practical use.

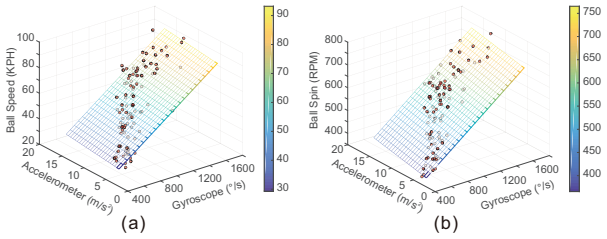

Fig. 5. Initialization of ball speed and spin.

In this paper, we measure ball state based on the integration of motion and visual sensing. It has two stages. First, we determine the roughly accurate ball state at the initial timestamp based on motion sensing of the player. Second, we adopt both ball state tracking methods mentioned above and combine their results optimally to produce accurate and robust tracking result. On one side, as visual positioning works independently at each timestamp, it is capable to cope with the noisy initial ball state and quickly rectify it. On the other side, ball motion physics-based calculation is robust to environmental noise. Next, we show the two stages of ball state measurement.

1) Initial Ball State Measurement: Initial ball state is determined by the impulse of the racket when it hits the ball. Fig. 6 shows the physical process. As the hit lasts a very short time [30], the force of the hit mostly decides the initial state. When the player waves the racket and hits the ball, motion sensors capture the force of the hit, which indicates the initial ball state. Fig. 5 shows our experimental results on initial ball state and motion sensing. The magnitude of initial ball state has positive correlation with accelerations and rotations of the player's movement. By using polynomial regression, the initial ball speed and spin can be roughly determined based on the player's motion sensor data. In addition, the orientation of the initial ball speed can be computed by rotations of the racket. The inaccuracy of initial state will be gradually reduced later by our tracking approach. Note that tracking ball state during its flight also requires the initial ball position. We obtain initial ball position by visual sensing. As position, speed, and spin (angular velocity) are three key variables to our tracking approach, we denote them at timestamp t as $\overrightarrow{P^t}$, $\overrightarrow{V^t}$, and $\overrightarrow{\omega^t}$, respectively.

2) EKF-based Ball State Tracking: We use both ball motion physics and visual positioning to track ball state. To obtain

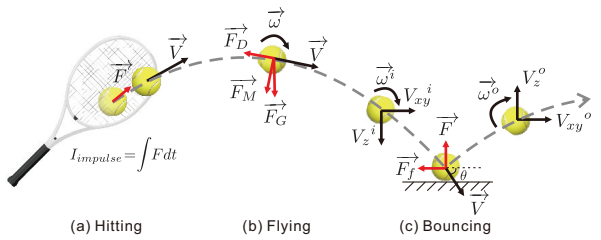

Fig. 6. Ball physical process: hitting, flying and bouncing.

a more accurate result based on the two rough independent measurements, we devise an EKF-based approach to optimally integrate the two measurements and produce a result that is closest to the "true" state. In the following, we first introduce the two tracking methods used in MV-Sports, and then explain how we combine their results to produce a better one.

The ball motion physics model requires the force analysis of the ball. In this paper, we present the basic analysis result based on aerodynamics studies whose details are given in [31], [32]. During tracking, there are two physical processes, flying and bouncing. When the ball is flying, it is affected by three forces, gravity ($\overrightarrow{F_G}$), air drag ($\overrightarrow{F_D}$) and Magnus force ($\overrightarrow{F_M}$). At timestamp t, they can be expressed as

$$\begin{cases} \left\|\overrightarrow{F_G}^t\right\| = mg \\ \overrightarrow{F_D}^t = -\frac{1}{2}\rho C_D A \left\|\overrightarrow{V}^t\right\| \overrightarrow{V}^t \\ \overrightarrow{F_M}^t = \frac{1}{2\pi}\rho C_L D^3 \overrightarrow{\omega}^t \times \overrightarrow{V}^t, \end{cases} \tag{3}$$

where $\|\cdot\|$ is the L^2-norm, m is the mass of the ball, g is the acceleration of gravity, ρ is the air density, C_D is the air drag coefficient, A is the effective cross-sectional ball area, C_L is the air lift coefficient, and D is the diameter of the ball. The values of these constants can be found in [31]. Based on Eq. (3), we can calculate the ball state iteratively. Denote the time period from $t-1$ to t as Δt and $\overrightarrow{F_1}^t = \overrightarrow{F_G}^t + \overrightarrow{F_D}^t + \overrightarrow{F_M}^t$. If Δt is small enough, say 16 ms in our implementation, $\overrightarrow{F_1}^t$ can be considered as constant. Then, we have

$$\overrightarrow{f}(\cdot) = \begin{cases} \overrightarrow{P^t} = \overrightarrow{P^{t-1}} + \overrightarrow{V^{t-1}}\Delta t + \frac{1}{2}\frac{\overrightarrow{F_1}^{t-1}}{m}\Delta t^2 \\ \overrightarrow{V^t} = \overrightarrow{V^{t-1}} + \frac{\overrightarrow{F_1}^{t-1}}{m}\Delta t \\ \overrightarrow{\omega^t} = \overrightarrow{\omega^{t-1}} \end{cases} \tag{4}$$

when the ball is in flight.

Fig. 6 illustrates the physical process of the ball bouncing on the ground. The friction between the ball and the ground as well as the elastic deformation of the ball are main reasons for the ball to lose horizontal and vertical kinetic energy, respectively [32]. As the ball bouncing takes very short time (usually less than 1 ms [32]), the change of ball speed and spin can be considered as instantaneous. Let $\overrightarrow{V} = [V_x \quad V_y \quad V_z]^T$ where V_j is the ball speed in the j-axis, and ${V_j}^i$ and ${V_j}^o$ are the ball speeds before and after bouncing, respectively. We apply similar notation to ball angular velocity ω and tangential velocity on the edge u. Vertically, we have

$${V_z}^o = C_b {V_z}^i, \tag{5}$$

where C_b is the bouncing coefficient. Horizontally, we define angle of incidence θ as the angle between ball speed and the ground. Namely, $\theta = \arctan\frac{{V_z}^i}{{V_{xy}}^i}$ and ${V_{xy}}^i = \sqrt{({V_x}^i)^2 + ({V_y}^i)^2}$. θ decides the type of friction between the ball and the ground. If it is lower than a threshold α, the friction is sliding. Otherwise, it is rolling friction. For sliding, we have

$$\begin{cases} {V_{xy}}^o = {V_{xy}}^i - C_f {V_z}^i (1 + C_b) \\ u^o = \frac{5(1+C_b)C_f}{2{V_z}^i} + u^i, \end{cases} \tag{6}$$

where C_f is the friction coefficient. For rolling, we have

$$\begin{cases} {V_{xy}}^o = \frac{5{V_{xy}}^i + 2u^i}{7} C_s \\ u^o = \frac{5{V_{xy}}^i + 2u^i}{7} C_t, \end{cases} \tag{7}$$

where C_s and C_t are the coefficients for the ball speed and the tangential velocity, respectively. Finally, we have

$$\begin{cases} {V_x}^o = \frac{{V_x}^i {V_{xy}}^o}{{V_{xy}}^i} \\ {V_y}^o = \frac{{V_y}^i {V_{xy}}^o}{{V_{xy}}^i} \\ \omega^o = \frac{u^o}{R}, \end{cases} \tag{8}$$

where R is the ball radius. C_b and C_f are both decided by the material of the ball and the surface of the ground. Their values, as well as the value of α, can be found in [32]. We empirically obtain the values of C_s and C_t in our experiments.

When using visual positioning technology to track the ball state, we first need to detect and localize the ball in video frames. Then, based on camera projection model and computer stereo vision theory [33], we can calculate the real-world position of the detected ball in MV-Sports' coordinate system (shown in Fig. 1). Due to varying lighting conditions and other practical factors, we cannot perfectly detect the ball in frames. Multiple "ball" candidates are usually detected containing the real ball as well as false positives. Besides, the calculated real-world positions of detected balls also have certain degree of error due to the motion blur caused by the ball's fast speed. These problems can be mitigated by our EKF-based tracking approach.

We call visual-positioning-based measurement observation, as it is obtained via cameras. We call ball-motion-physics-based measurement prediction because it is purely based on mathematical deduction. At every point in time, we apply prediction to obtain a roughly accurate position of the ball. It can identify the false positives of observation based on the distance between predicted and observed positions. Then, we employ the EKF to combine prediction and observation to produce more accurate ball state. The accumulated error of prediction can also be reduced.

Algorithm 2 shows our EKF-based approach to track ball state when the ball is flying. We formally define $\overrightarrow{S^t} = [\overrightarrow{P^t} \quad \overrightarrow{V^t} \quad \overrightarrow{\omega^t}]^T$ as the ball state, and $\overrightarrow{O^t}$ as the observation result of ball's position, at timestamp t. $\overrightarrow{S^t}$ is updated every timestamp. We also keep a posteriori error covariance matrix Sgm^t during the tracking. The basic steps of our approach are

Algorithm 2 EKF-based Tracking at Timestamp t

1: $\overrightarrow{S^{t|t-1}} \leftarrow \overrightarrow{f}(\overrightarrow{S^{t-1}})$ based on Eq. (4)
2: $Sgm^{t|t-1} \leftarrow GSgm^{t-1}G^T + R$
3: $K \leftarrow Sgm^{t|t-1}H^T(HSgm^{t|t-1}H^T + Q)^{-1}$
4: $\overrightarrow{S^t} \leftarrow \overrightarrow{S^{t|t-1}} + K(\overrightarrow{O^t} - H\overrightarrow{S^{t|t-1}})$
5: $Sgm^t \leftarrow (I - KH)Sgm^{t|t-1}$
6: **return** $\overrightarrow{S^t}, Sgm^t$

as follows. First, we apply prediction to compute intermediate variables $\overrightarrow{S^{t|t-1}}$ and $Sgm^{t|t-1}$, where G is the Jacobian matrix of $\overrightarrow{f}(\cdot)$ and R is a diagonal matrix with empirical errors of prediction on each element of $\overrightarrow{S^{t-1}}$. Second, we compute the gain matrix of observation K, where

$$H(i,j) = \begin{cases} 1 & if\ i = j \wedge i \in [1,3] \\ 0 & otherwise \end{cases}, \tag{9}$$

and Q is a diagonal matrix with empirical errors of observation on each element of $\overrightarrow{O^t}$. Third, we use K to adjust the prediction result $\overrightarrow{S^{t|t-1}}$ based on O^t and the intermediate covariance matrix $Sgm^{t|t-1}$, where I is the identity matrix. When the ball bounces on the ground, we simply apply Eq. (5)-(8) to update $\overrightarrow{S^t}$ as bouncing happens instantly. We also reset Sgm^t to the zero matrix to restart new iterations on Algorithm 2.

IV. Implementation and Evaluation

In this section, we present our system implementation, report our experimental results on our MV-Sports system, and show the system's performance.

A. Implementation

The implementation has three main parts: a motion part that collects and transmits motion sensor data of the player, a vision part that shoots the entire arena and transmits the video frames, and a back-end that receives M and V data and processes them.

As shown in Fig. 7, we use a wearable motion tracking device purchased from mbientlab [34] to capture motion data. It is placed on the player's wrist and provides 10 axes of motion sensing (3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, and altimeter/barometer/pressure). In this work, we only use the accelerometer and gyroscope data. Through the SDK provided by mbientlab, we program the device to capture 100 data samples per second and send the data to the back-end through Bluetooth. For vision part, we use two cheap USB cameras to collect video frames. We set the resolution as 1280 $\times$ 720 pixels and the frame rate as 60 fps. Both cameras are calibrated beforehand in the tennis arena [35]. We use a commodity laptop with one NVIDIA graphics card as the back-end. It synchronizes and processes the M and V data streams. The GPU is used for speeding up the image processing and neural network-based action recognition.

(a) Camera sensor (b) Wearable motion sensor

Fig. 7. Vision sensor and motion sensor.

In order to validate the ball state measured by our system, we need the ground truth ball state. We use a Bushnell-101921 velocity speed gun with 1 kph (0.28m/s) accuracy to measure ball speed. To measure ball spin, we use a high-end camera that captures the ball in flight with a frame rate of 240 fps.

B. Evaluation

We first evaluate the performance of player action recognition. Next, we evaluate the ball state measurement, including ball position, speed, and spin. We present the results on two metrics: accuracy and time cost.

1) Player Action Recognition: We collect samples for 5 typical tennis shots (i.e., forehand topspin, forehand slice, backhand topspin, backhand slice and serve) from 10 volunteers. Participants include 1 female and 9 males with ages ranging from 17 to 49 years old and heights ranging from 158 cm to 182 cm. Two of them are professional athletes and others are amateurs. We let each player freely play with a pitching machine and collect 50 samples for each action. Finally, with our highlighted action segmentation algorithm, we collect a large dataset including 2,500 video clips and corresponding motion streams. For samples collected for a specific shot from a specific person, we randomly pick 60% of them for training and use the rest for testing. We repeat such process for 10 times and use the mean of the results as the output.

Accuracy of Overall Performance: Fig. 8 shows confusion matrix for 5 actions according to overall classification accuracy. As we can see, MV-Sports achieves an average accuracy of 98.01% on player action recognition. For forehand topspin, backhand slice, and serve, MV-Sports achieves an average accuracy of 100%. We also observe that the accuracy is lower for backhand topspin. After investigating the reason behind this lower accuracy, we find that backhand topspin and backhand slice have inconspicuous inter-class differences that result in 7% topspin samples are incorrectly classified as slice.

Accuracy of Unseen User: To measure the accuracy of MV-Sports on unseen users, for each volunteer in our dataset, we test all samples collected from that volunteer using classification models generated from samples of all other volunteers. As shown in Fig. 9, MV-Sports achieves an average accuracy of 91.20% for actions performed by users it has not been trained on. We can see that except one person, the accuracies for unseen volunteers are all above 80% with small deviations. This shows that MV-Sports is robust and unaffected by unseen users.

Accuracy of Different Training Size: We evaluate the influence of the training data size by changing the percentage of data assigned for training. We compare MV-Sports with the

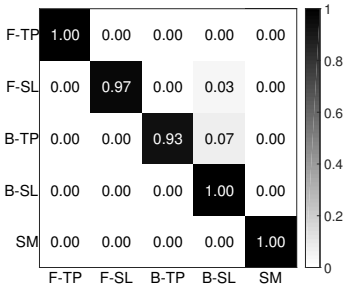

Fig. 8. Confusion matrix of player action recognition.

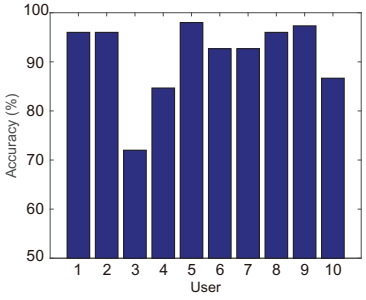

Fig. 9. Accuracy of unseen users.

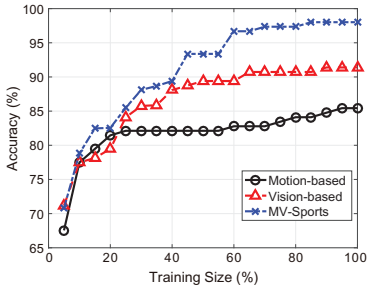

Fig. 10. Accuracy of different training data size.

Procedure	MV-Sports (ms/frame)
Highlight Segmentation	0.2
Player Localization	5.0
Feature Extraction	16.2
Online Action Classification	1.2
Sum	22.6

TABLE I
AVERAGE PROCESSING TIME OF EACH FRAME IN DIFFERENT STAGES OF ACTION RECOGNITION

motion-based and vision-based methods. Fig. 10 shows how they perform across different training data sizes. The results validate that our framework outperforms the other two on all training sizes. The motion-based and vision-based methods can achieve comparable accuracy with MV-Sports when the training data size is small, while MV-Sports yields better results when given sufficient training data.

Processing Time: Table I lists average processing time for different stages of action recognition. Added with the player localization time (5 ms) and the feature extraction time (16.2 ms), it takes 22.6 ms to process one frame. This means that the proposed method can process the sports videos and motion signals in real-time.

2) Ball State Measurement: We compare MV-Sports with a pure vision-based approach that visually localizes the ball using the same cameras as MV-Sports. The pure vision-based approach calculates the ball speed at each point in time by dividing the ball's displacement between two nearest positions over their time duration. However, it cannot calculate ball spin. The error is defined as the average difference between the evaluated approach and the ground truth. The processing time is the average time cost of processing at each time point. We collect 100 ball trajectories with different initial speeds and spins.

Accuracy of Ball Position: Fig. 11 shows the ball trajectory measured by our system and the vision-based approach. We can see that some positioning results of the vision-based approach are highly unstable. Namely, some positions are unusual and the curve they form does not follow aerodynamics. Such error mainly comes from the light condition and fast movements of the ball. MV-Sports, on the other hand, produces reliable ball positions during the whole flight. Using EKF-based approach that integrates MV data for ball tracking, our system can rectify the unstable visual positioning results. Thus, the trajectory obtained from MV-Sports is smoother,

Procedure	MV-Sports (ms/frame)	Vision-based (ms/frame)
Processing Time	10.3	10

TABLE II
AVERAGE PROCESSING TIME OF EACH FRAME OF BALL STATE MEASUREMENT

which indicates that the ball positions of MV-Sports are more accurate and reliable.

Accuracy of Ball Speed: Fig. 12 illustrates the speed error distribution of two tracking approaches. The median error for MV-Sports is around 0.5 m/s. It shows the high accuracy of our system as the median error is close to the error of our ground truth tool (0.28 m/s). Furthermore, our system is robust to practical factors as the error is lower than 1.5 m/s for over 90% of the trajectories. The pure vision-based approach, however, has much higher speed errors and a wider error distribution due to unstable positioning results. For 90% of the trajectories, its error is only lower than 2.5 m/s.

Accuracy of Ball Spin: Fig. 13 reports the CDF for spin error distribution of MV-Sports. The median error of our system is 40 RPM, which means the measurement is accurate as the normal ball spin is up to 1000 RPM. There are a few trajectories where our system cannot measure ball spin in a satisfying accuracy. It is mainly caused by the subtleness of ball spin. Unlike intrusive approaches like iBall [7] that embeds a motion sensor inside the ball, our system aims to be non-intrusive and practical. Although we cannot directly sense the ball spin, which raises certain inaccuracy, our system can still achieve reasonable performance for normal use.

Processing Time: Table II shows the average processing time for the two approaches at each time point. MV-Sports is 0.3 ms slower than the pure vision-based ball tracking. However, such difference is very minor and the overall time cost for MV-Sports is small enough to achieve real-time analysis.

V. CONCLUSION

In this paper, we presented MV-Sports, a cost-effective system for real-time sports analysis based on motion and vision sensor integration. As a case study for tennis, we aimed to accurately recognize the player actions and measure the ball state. For fine-grained action recognition, we leveraged motion signal to assist action highlighting and proposed a LSTM-based framework to integrate MV data for training and classification. For ball state measurement, we obtained

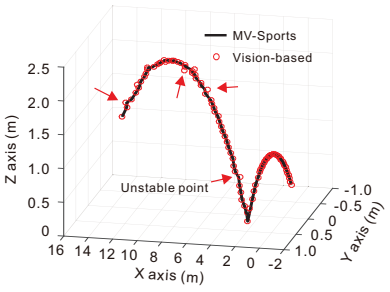

Fig. 11. Ball position tracking.

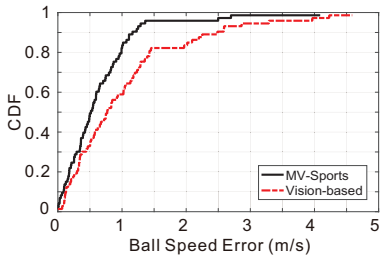

Fig. 12. CDF of ball speed error.

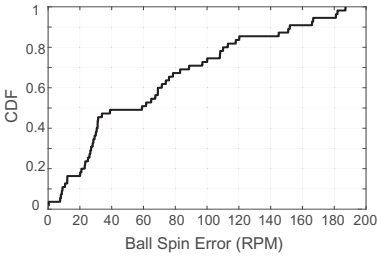

Fig. 13. CDF of ball spin error.

the initial ball state via motion sensing and devised an EKF-based approach to combine ball motion physics-based tracking and vision positioning-based tracking for more accurate ball state measurement. We implemented MV-Sports on COTS devices and conducted real-world experiments to evaluate the performance of our system. The results showed our approach can achieve accurate player action recognition and ball state measurement with sub-second latency. In the future, we will extend our system to other sports applications such as baseball and badminton.

ACKNOWLEDGMENTS

We sincerely thank anonymous reviewers and the area chair for their valuable comments. We acknowledge the support of Adam Champion, Ning Li, Zhiwei Shi, Wenjie Wang and Amy Xuan to our platform development and data collection.

REFERENCES

[1] D. S. Valter, C. Adam, M. Barry, and C. Marco, "Validation of prozone®: A new video-based performance analysis system," *International Journal of Performance Analysis in Sport*, vol. 6, no. 1, pp. 108–119, 2006.
[2] Vizrt, "Vizrt Sports." http://www.vizrt.com/sports/.
[3] O. Birbach, U. Frese, and B. Bäuml, "Realtime perception for catching a flying ball with a mobile humanoid," in *IEEE ICRA*, pp. 5955–5962, 2011.
[4] P. Halvorsen, S. Saegrov, A. Mortensen, D. K. C. Kristensen, A. Eichhorn, M. Stenhaug, S. Dahl, H. K. Stensland, V. R. Gaddam, C. Griwodz, and D. Johansen, "Bagadus: an integrated system for arena sports analytics: a soccer case study," in *ACM MMSys*, pp. 48–59, 2013.
[5] W. Liu, C. C. Yan, J. Liu, and H. Ma, "Deep learning based basketball video analysis for intelligent arena application," *Multimedia Tools Appl.*, vol. 76, no. 23, pp. 24983–25001, 2017.
[6] N. Owens, C. Harris, and C. Stennett, "Hawk-eye tennis system," in *International Conference on Visual Information Engineering*, pp. 182–185, 2003.
[7] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, L. Yang, S. Golwalkar, and A. Essanian, "Bringing IoT to sports analytics," in *USENIX NSDI*, pp. 499–513, 2017.
[8] G. Li, J. Teng, F. Yang, A. C. Champion, D. Xuan, H. Luan, and Y. F. Zheng, "EV-sounding: A visual assisted electronic channel sounding system," in *IEEE INFOCOM*, pp. 1483–1491, 2014.
[9] G. Li, F. Yang, G. Chen, Q. Zhai, X. Li, J. Teng, J. Zhu, D. Xuan, B. Chen, and W. Zhao, "EV-matching: Bridging large visual data and electronic data for efficient surveillance," in *IEEE ICDCS*, pp. 689–698, 2017.
[10] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "Trajectory-based ball detection and tracking in broadcast soccer video," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1164–1178, 2006.
[11] H. Chen, M. Tien, Y. Chen, W. Tsai, and S. Lee, "Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video," *J. Visual Communication and Image Representation*, vol. 20, no. 3, pp. 204–216, 2009.
[12] Babolat, "Babolat Play." http://en.babolatplay.com/.
[13] SONY, "Sony Tennis Sensor." http://www.sony.com.au/microsite/tennis/.
[14] ZEPP, "ZEPP." https://www.zepp.com/en-us/.
[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *IEEE CVPR*, pp. 1725–1732, 2014.
[16] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *IEEE ICCV*, pp. 4489–4497, 2015.
[17] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, pp. 568–576, 2014.
[18] C. Zhang, W. Liu, H. Ma, and H. Fu, "Siamese neural network based gait recognition for human identification," in *IEEE ICASSP*, pp. 2832–2836, 2016.
[19] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 11:1–11:28, 2017.
[20] D. Ravì, C. Wong, B. Lo, and G. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomedical and Health Informatics*, vol. 21, no. 1, pp. 56–64, 2017.
[21] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.
[22] F. Yan, W. Christmas, and J. Kittler, "A tennis ball tracking algorithm for automatic annotation of tennis match," in *BMVC*, vol. 2, pp. 619–628, 2005.
[23] X. Tong, J. Liu, T. Wang, and Y. Zhang, "Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video," *ACM Trans. Intelligent Systems and Technology*, vol. 2, no. 2, pp. 15:1–15:32, 2011.
[24] S. Tamaki and H. Saito, "Reconstruction of 3D trajectories for performance analysis in table tennis," in *IEEE CVPRW*, pp. 1019–1026, 2013.
[25] P. Dollár, R. Appel, S. J. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014.
[26] Q. Zhai, S. Ding, X. Li, F. Yang, J. Teng, J. Zhu, D. Xuan, Y. F. Zheng, and W. Zhao, "VM-tracking: Visual-motion sensing integration for real-time human tracking," in *IEEE INFOCOM*, pp. 711–719, 2015.
[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia*, pp. 675–678, 2014.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105, 2012.
[29] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv:1410.4615*, 2014.
[30] H. Brody, "Physics of the tennis racket," *American Journal of physics*, vol. 47, no. 6, pp. 482–487, 1979.
[31] Z. Zhang, D. Xu, and M. Tan, "Visual measurement and prediction of ball trajectory for table tennis robot," *IEEE Trans. Instrumentation and Measurement*, vol. 59, no. 12, pp. 3195–3205, 2010.
[32] R. Cross, "The bounce of a ball," *American Journal of Physics*, vol. 67, no. 3, pp. 222–227, 1999.
[33] P. F. Sturm, "Pinhole camera model," in *Computer Vision, A Reference Guide*, pp. 610–613, 2014.
[34] Mbientlab, "Mbientlab." https://mbientlab.com/.
[35] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.