TennisEye: Tennis Ball Speed Estimation using a Racket-mounted Motion Sensor

Hongyang Zhao
College of William and Mary
Williamsburg, VA
hyzhao@cs.wm.edu

Shuangquan Wang
College of William and Mary
Williamsburg, VA
swang10@email.wm.edu

Gang Zhou
College of William and Mary
Williamsburg, VA
gzhou@cs.wm.edu

Woosub Jung
College of William and Mary
Williamsburg, VA
wjung01@email.wm.edu

ABSTRACT

Aggressive tennis shots with high ball speed are the key factor in winning a tennis match. Today's tennis players are increasingly focused on improving ball speed. As a result, in recent tennis tournaments, records of tennis shot speeds are broken again and again. The traditional method for calculating the tennis ball speed uses multiple high-speed cameras and computer vision technology. This method is very expensive and hard to set up. Another way to calculate the tennis ball speed is to use motion sensors, which are lower cost and easier to set up. In this paper, we propose an approach for tennis ball speed estimation based on a racket-mounted motion sensor. We divide the tennis strokes into three categories: serve, groundstroke, and volley. For a serve, a regression model is proposed to estimate the ball speed. For a groundstroke or volley, two models are proposed: a regression model and a physical model. We use the physical model to estimate the ball speed for advanced players and the regression model for beginner players. Under the leave-one-subject-out cross-validation test, evaluation results show that TennisEye is 10.8% more accurate than the state-of-the-art work.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous computing**; **Mobile computing**; **Mobile devices**;

KEYWORDS

tennis ball speed estimation, motion sensor, regression model, physical model

ACM Reference Format:
Hongyang Zhao, Shuangquan Wang, Gang Zhou, and Woosub Jung. 2019. TennisEye: Tennis Ball Speed Estimation using a Racket-mounted Motion Sensor. In *Proceedings of the 18th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN'19).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

IPSN'19, April 2019, Montreal, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the advance of ubiquitous computing, cyber physical systems, and human computer interaction, wearable devices are becoming more and more popular nowadays [18] [9]. Applications of wearable devices have been widely extended to the field of sports. For example, Hao et al. propose a running rhythm monitoring system based on the sound of breathing through smartphone embedded sensors [12]. Kranz et al. propose an automated assessment system for balance board training called Gymskill, which provides feedback on training quality to the user based on smartphone integrated sensors [19]. In addition, there are several studies in other sports like running [1], skiing [10], climbing [15], cricket [14], football [44], and table tennis [3]. In addition to these research publications, the industrial wearable devices market is also evolving at a rapid pace. The global industrial wearable devices market has reached a value of 1.5 billion dollars in 2017, and is expected to have a compound annual growth rate (CAGR) of 9.6% [16]. The wearable devices market in sports is expected to register a CAGR of 9.8%, during the forecast period (2018 - 2023) [13].

There are also applications of wearable devices in tennis training. Several commercial tennis assistant systems are available on the market that aim to improve players' performance [41] [32] [24] [7]. These products either integrate the motion sensors inside the racket [7], or require users to attach the motion sensors to the racket [41] [32] [24]. They analyze the motion sensor data and compute the key performance metrics for each swing, such as stroke type, ball speed, ball spin, and ball impact location. In addition to these commercial products, there are also several existing research works on analyzing the performance of tennis shots. For instance, Srivastava et al. analyze the consistency of the tennis shots [31]. The authors provide recommendation on wrist rotation based on the shots from professional players. Sharma et al. analyze the tennis serve [28]. By comparing the serve phases of a user to those of professionals, the system provide the user with corrective feedback and insights into their playing styles.

Tennis ball speed is an important metric in assessing the skill level of a tennis player. There are two main ways to calculate the tennis ball speed. One way is to use multiple high-speed cameras to capture ball movement and calculate speed, such as Hawk-eye technology [8] [21] and PlaySight [25]. These systems use advanced

This work is supported by U.S. National Science Foundation under grants CNS-1253506 (CAREER).

image processing and analytical algorithms to capture ball movement and calculate speed. However, high-speed cameras are very expensive and hard to set up. Therefore, most players cannot get access to these systems, which limits their popularity. Another way is to use motion sensors. Compared with camera-based method, the motion sensors-based method is lower cost, more energy efficient, not influenced by lighting environment, and easier to set up. There are some commercial products on the market that assess the performance of the players and estimate the ball speed [41] [32] [24] [7]. However, none of these commercial products open their algorithms to the public. In addition, to our knowledge, no previous publication has used motion sensors to calculate the tennis ball speed. Therefore, we are motivated to explore how to use a motion sensor to calculate the tennis ball speed.

There are several research publications and commercial products that use motion sensors to analyze tennis shots. However, none of them open their source codes and datasets to the public. Source codes and tennis dataset sharing are valuable as they allow researchers to build their works upon others rather than repeat already existing research work. In addition, they encourage more connection and collaboration between researchers, which promotes the tennis research. Therefore, we are motivated to open our source codes and tennis dataset to the public[1].

We present TennisEye, a tennis ball speed calculation system using a racket-mounted motion sensor. We apply a simple threshold-based method to detect if there is a tennis stroke. Once a tennis stroke is detected, we use a time window to extract stroke data and interpolate the sensor readings if the true value exceeds the measurement limit. We apply the Random Forest classifier to classify each tennis stroke into one of three stroke types: serve, groundstroke, and volley. A serve is a shot to start a point. A groundstroke is a shot that is executed after the ball bounces once on the ground, while a volley is a shot that is executed before the ball bounces on the ground. If the tennis stroke is a serve, we apply a regression model to calculate serve speed. If the tennis stroke is a groundstroke or volley, we propose two models: a physical model and a regression model. For advanced players, they have correct and constant stroke gestures. We use the physical model to calculate the ball speed for them. For beginner players, they have incorrect and varying stroke gestures. We use the regression model to calculate the ball speed.

We summarize our contributions as follows:

(1) We propose a tennis ball speed calculation system, TennisEye. It is the first research publication to calculate the serve, groundstroke and volley speed of a tennis ball using a racket-mounted motion sensor.
(2) We propose two models to calculate the groundstroke or volley speed: a physical model and a regression model. We apply the physical model to calculate the ball speed for advanced players, and the regression model to calculate the ball speed for beginner players.
(3) We evaluate the proposed system using the tennis shot data from players of different levels. Our experiment results show that TennisEye is 10.8% more accurate than the state-of-the-art work.

[1]https://hongyang-zhao.github.io/TennisEye/

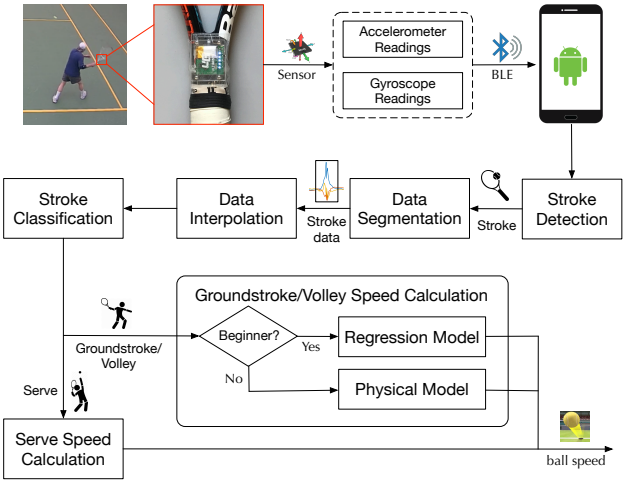

Figure 1: Data Processing of TennisEye

(4) We collect an accurate and high-quality tennis dataset. We open our source codes and tennis dataset to the public.

The remainder of this paper is organized as follows. First, we introduce the system design in Section 2. In Section 3, we evaluate the system performance. We discuss the system and future work in Section 4. We summarize the related works in Section 5. Finally, we draw our conclusion in Section 6.

2 TENNISEYE DESIGN

We introduce the system design of TennisEye in this section. First, we introduce the data processing in Section 2.1. Then, we introduce the sensor deployment and data collection in Section 2.2. Following that, modules in the data processing are introduced from Section 2.3 to Section 2.8.

2.1 Overview of TennisEye

The data processing of TennisEye is shown in Fig. 1. A motion sensor is deployed on a racket handle to collect the accelerometer and gyroscope readings, which are real-time transmitted to a smart phone through Bluetooth Low Energy (BLE). Based on the collected motion data, a threshold-based method is proposed to detect tennis strokes. Once a tennis stroke is detected, we apply a peak detection algorithm to find the impact time when a ball hits the racket. Once the impact time is found, we use a sliding window with a length of 2 seconds to extract the stroke data. For the stroke data, we apply a cubic spline interpolation to compensate the sensor readings if the sensor saturates. Then, we use the Random Forest classifier to classify each tennis stroke into one of three stroke types: serve, groundstroke, and volley. If the tennis stroke is a serve, we propose a serve speed estimation method to calculate serve speed. If the tennis stroke is a groundstroke or volley, we propose two models, a regression model and a physical model, to calculate the ball speed for the beginner player and the advanced player, respectively. We train the models and evaluate the performance of the system using a laptop.

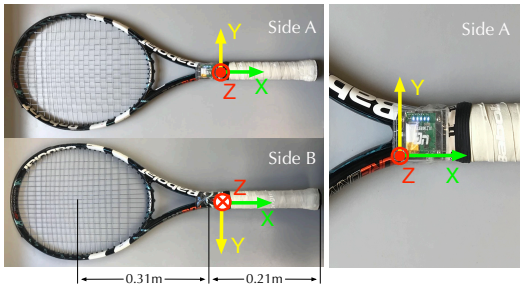

Figure 2: Sensor placement and coordinate system

Figure 3: Birdview of the tennis court from a PlaySight camera

2.2 Sensor Deployment

The motion sensor we used is a UG sensor [43]. It includes a triaxial acceleration sensor and a triaxial gyroscope. The measure range of the accelerometer and the gyroscope were set to be $\pm$ 16g and $\pm$ 2000$^{\circ}/sec$, respectively. Both sensors were sampled with 100 Hz. The UG sensor was fixed at the handle of the players' rackets. Fig. 2 shows the sensor position and the coordinate system. We call the side with the UG sensor side A, and the side without the UG sensor side B.

We collected data in a tennis court that was equipped with a PlaySight system [25], which included six high definition (HD) cameras. The PlaySight system uses image processing algorithm to recognize stroke type, ball speed, and more. We use the stroke type and the ball speed calculated by the PlaySight system as the ground truth. The birdview of the tennis court from a PlaySight camera is shown in Fig. 3.

2.3 Stroke Detection

When a player plays tennis, he/she not only swings a racket, but also performs some non-stroke actions during a match. For example, he/she may run on the court, use the racket to pick up a ball from the court surface, or twirl the racket in his/her hands while waiting for an opponent to serve. The aim of the stroke detection is to detect stroke behaviors, rather than those non-stroke actions.

We find that when a player hits a ball, the player swings the racket in a circular motion. In this circular motion, the racket is affected by a centripetal acceleration that points to the human torso. This centripetal acceleration generates a spike in the accelerometer readings in X-axis, a_x. Fig. 4 shows a_x when a player plays tennis.

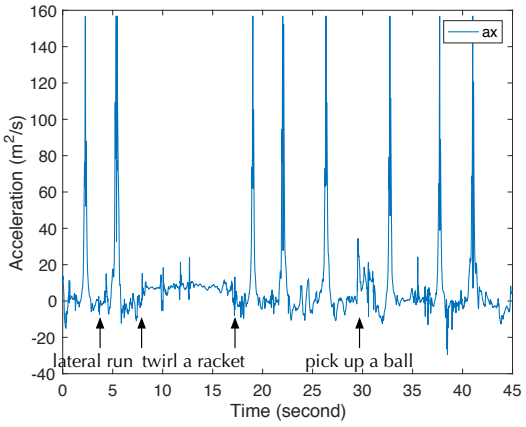

Figure 4: Accelerometer readings in the X-axis when a player plays tennis

Each spike in the figure is a tennis stroke. Other non-stroke actions, such as lateral run, twirling a racket, or picking up a ball, do not generate large values in a_x. Therefore, we use a_x to detect stroke behaviors. If a_x is larger than a threshold, i.e. 9g, it is regarded as a stroke.

2.4 Data Segmentation

After a stroke is detected, the next step is to find the start and end time of a tennis stroke and extract this interval of sensor data as stroke data. The extracted stroke data will be used for ball speed calculation.

There are mainly four phases for a tennis stroke: backswing, acceleration, impact, and follow-through. In a backswing phase, a player moves a racket behind his/her body and prepares to swing forward. In an acceleration phase, the player moves the racket forward. The speed of the racket increases in this phase. At the end of the acceleration phase and the start of the impact phase, the speed of the racket reaches maximum. In an impact phase, the racket hits a tennis ball. Due to impact, the speed of the racket decreases rapidly. Finally, in a follow-through phase, the player slows down the racket after hitting a ball.

Based on our study, we find that the impact phase occurs in the middle of these four phases. In addition, the speed of the racket reaches maximum at the start of the impact phase. Therefore, by looking for the time when the speed of the racket reaches maximum, we find the start of the impact phase. In our coordinate system, no matter which stroke (serve, forehand, backhand) a player plays, the racket always rotates around Y-axis. Therefore, the position with the max absolute value of the gyroscope readings in the Y-axis is the start of the impact phase.

Fig. 5 shows the gyroscope readings of a tennis serve. From the figure, we find that the valley of g_y is the start of the impact phase. We apply a peak detection algorithm with a sliding window on the absolute values of the gyroscope readings in the Y-axis, $|g_y|$, to find the impact start time. As the minimum interval between two consecutive strokes is 2 seconds, the time window size is set to be 2 seconds. Specifically, $|g_y(t)|$ is a peak if it is larger than all of the samples in the time window of $[t - 1s, t + 1s]$. Since the

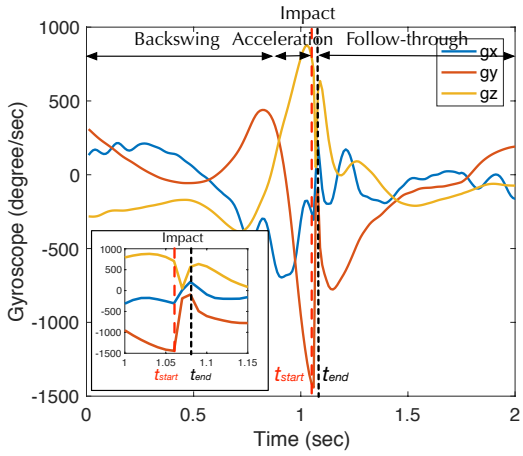

Figure 5: Gyroscope readings of a serve

minimum of $|g_y|$ for a stroke in our dataset is 814.5 degree/sec, the detected peak of $|g_y|$ should be larger than 814.5 degree/sec. After the impact start time t_{start} is detected, we use a time window of $[t_{start} - 1s, t_{start} + 1s]$ to extract the stroke data.

2.5 Data Interpolation

In our dataset, we find that the gyroscope readings in the Y-axis sometimes saturate when a player serves with a high speed. This saturation happens when a sensor measures a value that is larger than its measurement range, $\pm\ 2000^{\circ}/sec$. The blue line in Fig. 6 shows an example of the sensor saturation. In the figure, we find that the raw gyroscope readings in the Y-axis saturate from 0.52 seconds to 0.54 seconds. To deal with this problem, we apply the cubic spline interpolation [6] to interpolate the saturated gyroscope readings in the Y-axis. Specifically, we use the gyroscope readings in the Y-axis within the range of $[t_{sat\_start} - tw, t_{sat\_start})$ and $(t_{sat\_end}, t_{sat\_end} + tw]$ to construct new data points within the range of $[t_{sat\_start}, t_{sat\_end}]$. t_{sat_start} and t_{sat_end} are the start and end times of the saturation, which are 0.52 seconds and 0.54 seconds in this case. tw is empirically set to be 100ms. The gyroscope readings in the Y-axis after interpolation are shown as the red dashed line in the figure.

2.6 Stroke Classification

After data interpolation, the next step is to classify the stroke type for each stroke data segment. First, we extract a series of features from each data segment. The features include mean, standard deviation, skewness, kurtosis, minimum, and maximum of each axis of accelerometer and gyroscope readings. The amplitude of accelerometer and gyroscope readings are also computed. In total, 38 features for each data segment are calculated and normalized between $[0, 1]$. Second, we apply a machine learning classifier to classify each data segment into one of three stroke types: serve, groundstroke, and volley. We compare the performance of five classifiers: Naive Bayes, AdaBoost, Support Vector Machine (SVM), Decision Tree, and Random Forest. Since the Random Forest performs best as shown in Section 3.4, we choose the Random Forest classifier for stroke classification.

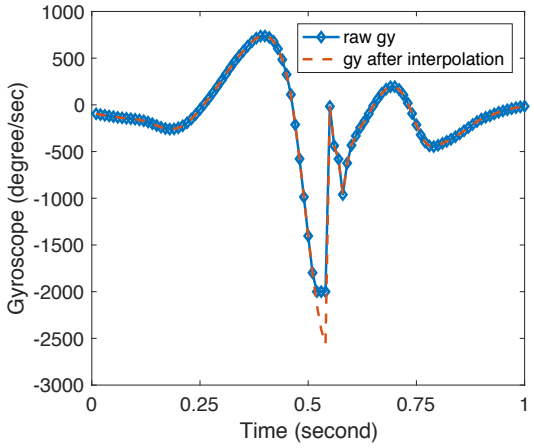

Figure 6: Interpolation of the gyroscope readings in the Y-axis

2.7 Serve Speed Calculation

For a tennis serve, the initial speed of a ball is quite small and mainly depends on the racket speed. The larger the racket speed, the higher the ball speed. This motivates us to use the racket speed to calculate the serve speed. When a player serves a tennis ball, the racket swings in a circular motion with the human shoulder as the center. Because the racket rotates around the Y-axis in our coordinate system, g_y measures the angular speed of the racket. We use g_y at the impact start time to model the ball serve speed v_{b_serve} by a linear regression model as:

$$v_{b\_serve} = k \cdot |g_y(t_{start})| + b, \tag{1}$$

where k and b are the parameters of the linear regression model. All of the tennis serve data are used to train the serve model. k and b are calculated by using the method of least squares [30]. k and b are 0.025 and 20.06, respectively.

2.8 Groundstroke/Volley Speed Calculation

We propose two models to calculate the ball speed: a physical model and a regression model. The physical model is built based on the physical impact between a racket and a ball, while the regression model applies a linear regression model to estimate the ball speed. For advanced players, they have correct stroke gestures, and similar gestures for the same type of the stroke. We propose a physical model to calculate the ball speed for them. For beginner players, it is hard to build the physical model due to two reasons. First, their stroke gestures are incorrect. With incorrect stroke gestures and terrible tennis skills, they often swing the racket in an awkward way, which is hard to build the physical model. Second, for the same type of the stroke, they may perform different gestures and swing the racket in different ways. The big variance in their stroke gestures reduces the accuracy of the physical model. Therefore, instead of building a physical model, we propose a simple regression model to calculate the ball speed for the beginner players.

2.8.1 Physical Model. The impact between a racket and a tennis ball is governed by some physical laws and mechanical principles. By applying these laws and principles, we propose a physical model

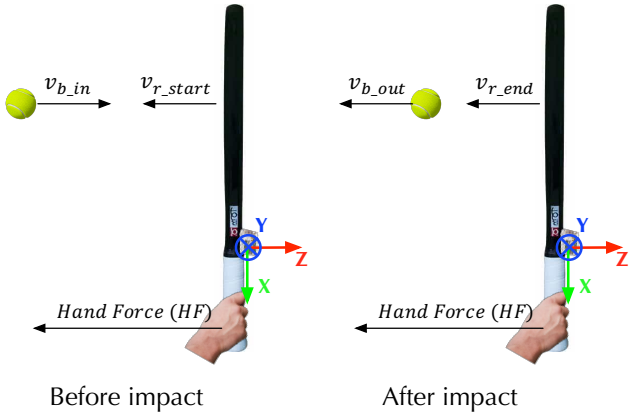

Figure 7: Physical impact process between a tennis racket and a tennis ball

to estimate ball speed. The outgoing ball speed is mainly influenced by two factors: the racket speed and the incoming ball speed. The racket speed can be calculated using motion sensors, while the incoming ball speed is unknown. Without the incoming ball speed, the outgoing ball speed cannot be calculated by a physical model. To solve this problem, we apply a physical law (conservation of linear momentum [20]) and a mechanical principle (Coefficient of Restitution [37]). Both the physical law and the mechanical principle contain the incoming ball speed parameter. We combine these into one by eliminating the incoming ball speed parameter to get the outgoing ball speed.

Because of the difficulty in characterizing the physical impact process using only one racket-mounted motion sensor, we make several assumptions to simplify the physical model.

(1) The tennis ball horizontally hits and bounces off the racket.
(2) The ball impacts at the center of the racket face.
(3) During impact, there is a constant hand force on the racket.

There are four reasons to these three assumptions. (1) The physical impact process between a racket and a tennis ball is very complicated. Without these assumptions, it is extremely difficult to build a physical model only using motion sensors. (2) Even if we build a model without these simplified assumptions, this model will be very complicated. A complicated model may require much more computation and energy cost than the simplified model. This is not feasible for the small-size device embedded into the racket. (3) As shown in Section 3, the evaluation results demonstrate that our proposed models perform quite well already. A complicated model may only increase the accuracy a little. (4) These assumptions are made based on our observations on tennis matches. We find that the angle between a ball and the normal to the racket string plane is typically quite small. Otherwise, the ball will either move towards the ground or move towards the sky. Accordingly, we assume that the ball horizontally hits and bounces off the racket. In addition, we find that tennis balls usually hit at the center of the rackets for advanced players. Therefore, we assume that the ball impacts at the center of the racket face. Finally, we find that the duration of the impact is roughly 20ms, as shown in Fig. 5. Thus, it is reasonable to assume a constant hand force during this short period of time.

Based on these assumptions, the physical impact process is shown in Fig. 7. Before impact, there is a hand force HF exerted on a racket. Under the influence of the hand force, the racket of mass M is moving at velocity v_{r_start} towards a tennis ball. At the same time, this tennis ball of mass m is moving at velocity v_{b_in} towards the racket. After impact, the tennis ball bounces off the racket at velocity v_{b_out}, reducing the velocity of the racket to v_{r_end}.

By conservation of linear momentum, we get:

$$\int_{t_{start}}^{t_{end}} HFdt = M\cdot\left(v_{r\_end} - v_{r\_start}\right) + m\cdot\left(v_{b\_out} + v_{b\_in}\right), \quad (2)$$

where t_{start} and t_{end} are the start and end times of impact. In this equation, t_{start}, M, and m are known. t_{start} is calculated in data segmentation module, as shown in Section 2.4. M and m are $300g$ and $50g$. To calculate v_{b_out}, we need to calculate the other unknown parameters first. They are t_{end}, $v_{r\_end} - v_{r\_start}$, HF, and v_{b_in}.

Calculating $\mathbf{t_{end}}$. The impact starts when a ball hits a racket, and ends when the ball leaves the racket. During the impact, due to the momentum lost by the ball, the speed of the racket decreases rapidly. After the ball leaves the racket, the speed of the racket increases again. By searching for the change of the racket speed, we find the end time of impact as shown in Fig. 8. More specifically, by searching from t_{start}, t_{end} is calculated as:

$$t_{end} = t \quad if\ \left|g_y\left(t\right)\right| < \left|g_y\left(t+1\right)\right| \quad (3)$$

Calculating $\mathbf{v_{r\_end} - v_{r\_start}}$. $v_{r\_end} - v_{r\_start}$ can be calculated as the integration of the accelerometer readings in the Z-axis. However, as there are two sides of the rackets, the accelerometer readings in the Z-axis will have opposite values when the tennis ball hits on the different sides. The motion sensor is attached on the side A of the racket, as shown in Fig. 2. When the side A of the racket hits the tennis ball, the movement direction of the racket is the same as the positive direction of the Z-axis of the sensor. $v_{r\_end} - v_{r\_start}$ is calculated as the integration of the accelerometer readings in the Z-axis. However, when the side B of the racket hits the tennis ball, the movement direction of the racket is opposite to the positive direction of the Z-axis of the sensor, as shown in Fig. 7. $v_{r\_end} - v_{r\_start}$ is calculated as the integration of the inverse of the accelerometer readings in the Z-axis. Therefore, to calculate the ball speed, we need to reorientate the motion sensor so that the positive direction of the Z-axis of the sensor is the same as the movement direction of the racket.

There are eight possible impact types between a tennis ball and a racket, which are summarized in Table 1. From the table, we find that when the $g_y(t_{start})$ is larger than 0, the positive direction of the Z-axis of the sensor is the same as the movement direction of the racket. When the $g_y(t_{start})$ is smaller than 0, the positive direction of the Z-axis of the sensor is opposite to the movement direction of the racket. This motivates us to use $g_y(t_{start})$ to reorientate the sensor readings in Z-axis. The reoriented accelerometer readings in the Z-axis $a_z^{'}$ is calculated as:

$$a_z^{'} = \begin{cases} a_z & g_y\left(t_{start}\right) > 0 \\ -a_z & g_y\left(t_{start}\right) < 0 \end{cases} \quad (4)$$

The change of the racket speed $v_{r\_end} - v_{r\_start}$ in Eq. 2 is calculated as the integration of the reoriented acceleration readings

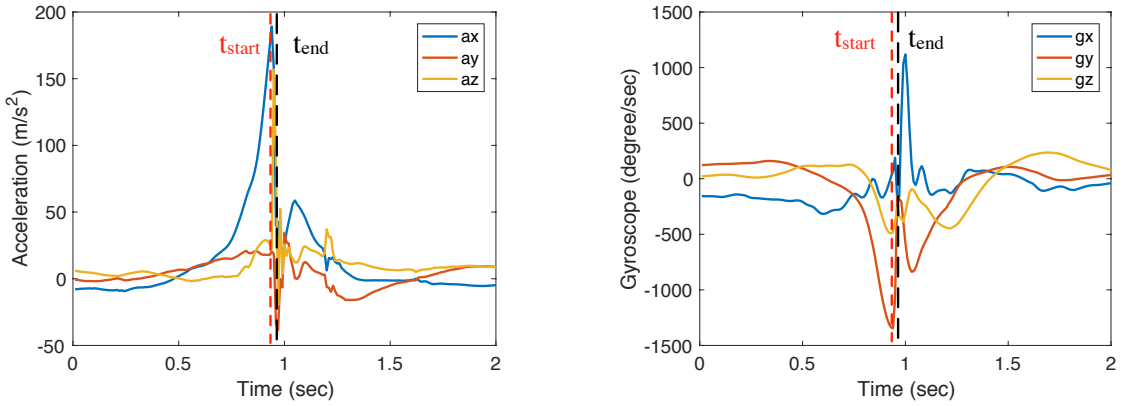

Figure 8: Acceleration data (left figure) and gyroscope data (right figure) of a forehand shot performed by a left-hand player. The start time of impact t_{start} is marked as the red dashed line.The end time of impact t_{end} is marked as the black dashed line.

Table 1: Eight impact types between a tennis ball and a racket

Dominant Hand	Forehand/Backhand	Side that a ball hits a racket	$g_y(t_{start})$	$Direction_{Z-axis}$ and $Direction_{racket}$
Left	Forehand	Side A	> 0	Same
	Forehand	Side B	< 0	Opposite
	Backhand	Side A	> 0	Same
	Backhand	Side B	< 0	Opposite
Right	Forehand	Side A	> 0	Same
	Forehand	Side B	< 0	Opposite
	Backhand	Side A	> 0	Same
	Backhand	Side B	< 0	Opposite

in the Z-axis during impact as:

$$v_{r\_end} - v_{r\_start} = \int_{t_{start}}^{t_{end}} a_z^{'}(t)\, dt \tag{5}$$

Calculating HF. This hand force causes the racket to move in an accelerated motion, which is measured by the Z-axis of the accelerometer. As we assume that the hand force is constant during impact, we use the adjusted accelerometer readings in the Z-axis $a_z^{'}$ at the start time of impact t_{start} to model the hand force. The larger the hand force, the larger the acceleration. Thus, we choose a linear function to model this relationship as shown below:

$$HF = k_{HF} \cdot M \cdot a_z^{'}\left(t_{start}\right) + b_{HF}, \tag{6}$$

where k_{HF} and b_{HF} are the model parameters. After substituting Eq. 3 through Eq. 6 into Eq. 2, Eq. 2 has four unknown parameters: $k_{HF}, b_{HF}, v_{b\_in}, v_{b\_out}$. We use the incoming ball speed v_{b_in} and outgoing ball speed v_{b_out} calculated from the PlaySight system to train this model. First, we get the incoming and outgoing ball speeds for each tennis shot from the PlaySight system. Then, we feed the incoming and outgoing ball speeds of all the tennis shots into Eq. 2. Finally, we use the method of least squares [30] to calculate k_{HF} and b_{HF}. k_{HF} and b_{HF} are 0.236 and 65.83, respectively.

Calculating v_{b_in}. After k_{HF} and b_{HF} are determined, there are two unknown parameters in Eq. 2: v_{b_in} and v_{b_out}. To calculate v_{b_out}, we need to calculate v_{b_in} first. However, v_{b_in} is hard to be calculated using a motion sensor. To solve this problem, we apply the coefficient of restitution (COR) to build another equation between v_{b_in} and v_{b_out}. We combine these two equations into one by eliminating v_{b_in} parameter to calculate v_{b_out}.

The COR is defined as the ratio of the final velocity to the initial velocity between two objects after their collision [37]. The COR is a measure of how much kinetic energy remains after the collision of two bodies, with the value that ranges from 0 to 1. If the COR is close to 1, it suggests that very little kinetic energy is lost during the collision; on the other hand, if it is close to 0, it indicates that a large amount of kinetic energy is converted into heat or otherwise absorbed through deformation. In our case, the COR is calculated as:

$$e = \frac{v_{b\_out} - v_{r\_end}}{v_{r\_start} + v_{b\_in}} \tag{7}$$

To calculate the outgoing ball speed v_{b_out}, we need to calculate v_{r_start}, v_{r_end}, e, and v_{b_in} first. The racket speed before impact v_{r_start} is calculated by the gyroscope readings in the Y-axis:

$$v_{r\_start} = \left|g_y\left(t_{start}\right)\right| \cdot R_{swing}, \tag{8}$$

where R_{swing} is the swing radius of the circular motion, which is the distance between the center of the racket face and the rotation center. Howard Brody measured the swing radius among a number of players [4]. He found that the average radius between the butt end of the racket and the rotation center is 0.2m. As the distance

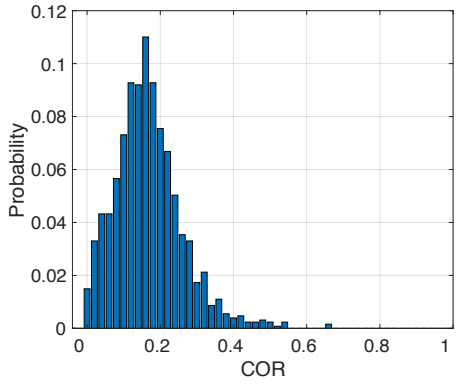

Figure 9: Distribution of COR values

between the center of the racket face and the butt end of the racket is 0.52m, as shown in Fig. 2. Therefore, R_{swing} is set to be 0.72m.

By substituting v_{r_start} into Eq. 5, we calculate the racket speed after impact v_{r_end} as:

$$v_{r\_end} = v_{r\_start} + \int_{t_{start}}^{t_{end}} a_z{}'(t)\, dt \tag{9}$$

We feed the incoming ball speeds and outgoing ball speeds calculated from the PlaySight system into Eq. 7 to calculate e. The distribution of all calculated COR values is shown in Fig. 9. We find these COR values obey a normal distribution with a mean of 0.16. Because there are many impact factors for COR estimation (such as ball speed, stroke force, string tension and impact position) and the correlations between these impact factors and the COR are unknown, it is extremely difficult to explicitly define the COR model. In addition, most COR values are close to the mean of this normal distribution. Therefore, we assume COR to be a constant value, 0.16, for all the tennis shots to simplify the model.

We then combine Eq. 9 and Eq. 2 by eliminating the incoming ball speed to get the outgoing ball speed as:

$$v_{b\_out} = \frac{e}{1+e} \cdot \frac{1}{m} \left[m \cdot v_{r\_start} - M \cdot \left(v_{r\_end} - v_{r\_start} \right) + \frac{m}{e} \cdot v_{r\_end} + \int_{t_{start}}^{t_{end}} HF dt \right] \tag{10}$$

From the above equation, we find that the outgoing ball speed v_{b_out} depends on v_{r_start}, v_{r_end}, t_{start}, t_{end}, and HF. The incoming ball speed v_{b_in} does not appear in the equation as it is eliminated when we combine Eq. 2 and Eq. 7. Though v_{b_in} does not appear in Eq. 10, it indirectly influences the outgoing ball speed via v_{r_end}, which is calculated based on the integration of acceleration data in the Z-axis during impact as shown in Eq. 9. The larger incoming ball speed, the larger fluctuation of the acceleration data during impact. Therefore, the influence of the incoming ball speed is considered in our model.

2.8.2 Regression Model. When a player performs a groundstroke or volley, he/she swings the racket in a circular motion. In our coordinate system, the racket rotates around the Y-axis for both strokes. Similar to the serve speed model, we use g_y to calculate the outgoing ball speed v_{b_out} by a linear polynomial model as:

$$v_{b\_out} = k \cdot \left| g_y\left(t_{start}\right) \right| + b, \tag{11}$$

where k and b are the parameters of the linear regression model. We use all the tennis groundstroke and volley data to train this model by the method of least squares [30]. k and b are 0.039 and 3.94, respectively.

3 PERFORMANCE EVALUATION

We evaluate the performance of TennisEye in this section. We first introduce the data set in Section 3.1. Then, we introduce the performance of the physical model and regression model using leave-one-subject-out cross validation, self test, and 5-fold cross validation in Section 3.2.1, Section 3.2.2, and Section 3.2.4, respectively. After that, we evaluate the performance of stroke detection in Section 3.3 and stroke classification in Section 3.4. Finally, we evaluate the overall performance of TennisEye in Section 3.5.

3.1 Data Set

We collected data from 7 players. Based on their self-report information, we divided the subjects into three categories: coach, regular player, and casual player. The coaches play several times per week, the regular players play one time per week, and the casual players play 0~2 times per month. Table 2 shows the summary of tennis data we collected with a UG sensor. In total, we collected 569 serves, 1398 groundstrokes, and 18 volleys.

No previous publication has used motion sensors to estimate tennis ball speed. However, we are aware that there are some tennis sensors on the market, such as Zepp [41], Sony Sensor [29], Usense [32], and Babolat Play [24]. These sensors are attached to the tennis racket. By analyzing the tennis data, they compute the key performance metrics for each swing, such as stroke type and ball speed. Among them, Sony Sensor and Usense are no longer manufactured. Babolat Play does not compute the ball speeds of groundstroke and volley shots, which we focus on. Zepp is a popular tennis sensor that recognizes stroke types and calculates serve, groundstroke and volley speeds. Therefore, we choose Zepp as state-of-the-art work. To compare with the ball speed calculation algorithm in Zepp, we collected tennis data using both a UG and Zepp sensor. Table 3 shows the summary of the tennis data collected.

3.2 Ball Speed Calculation Accuracy

3.2.1 Leave-one-subject-out Evaluation. To evaluate the performance of the proposed ball speed models. First, we use the video data to locate each tennis stroke in the dataset collected with a UG sensor as shown in Table 2. Then, we calculate the tennis ball speeds for each stroke based on the proposed ball speed models. Finally, we compare the ball speeds calculated by our model with the ball speeds calculated by other models. Here, we use the leave-one-subject-out cross validation to evaluate the performance of the proposed ball speed models. The leave-one-subject-out cross validation uses the tennis shot data from 6 subjects to train the ball speed estimation model, and then applies this model to estimate the ball speed from the remaining subject. Mean and standard deviation of error are considered as the evaluation metrics. The results are shown in Fig. 10. For the serve ball speed estimation,

Table 2: Dataset collected with a UG sensor

Player	Player Description	Dominant Hand	Data Collection Time	# Serve	# Groundstroke	# Volley
# 1	Male Coach	Right	5/8/2018	36	0	0
# 2	Female Coach	Left	5/15/2018; 5/22/2018	75	214	5
# 3	Female Coach	Right	8/13/2018	50	136	0
# 4	Regular Player	Right	5/23/2018	0	302	3
# 5	Casual Player	Left	7/7/2018; 7/15/2018; 7/22/2018	0	628	10
# 6	Casual Player	Right	5/7/2018; 8/13/2018; 9/7/2018	263	118	0
# 7	Casual Player	Right	9/7/2018	145	0	0

Table 3: Dataset collected with both a UG and a Zepp sensor

Player	Data Collection Time	# Serve	# Groundstroke	# Volley
# 3	8/13/2018	50	136	0
# 5	7/22/2018	0	209	6
# 6	9/7/2018	46	118	0
# 7	9/7/2018	145	0	0

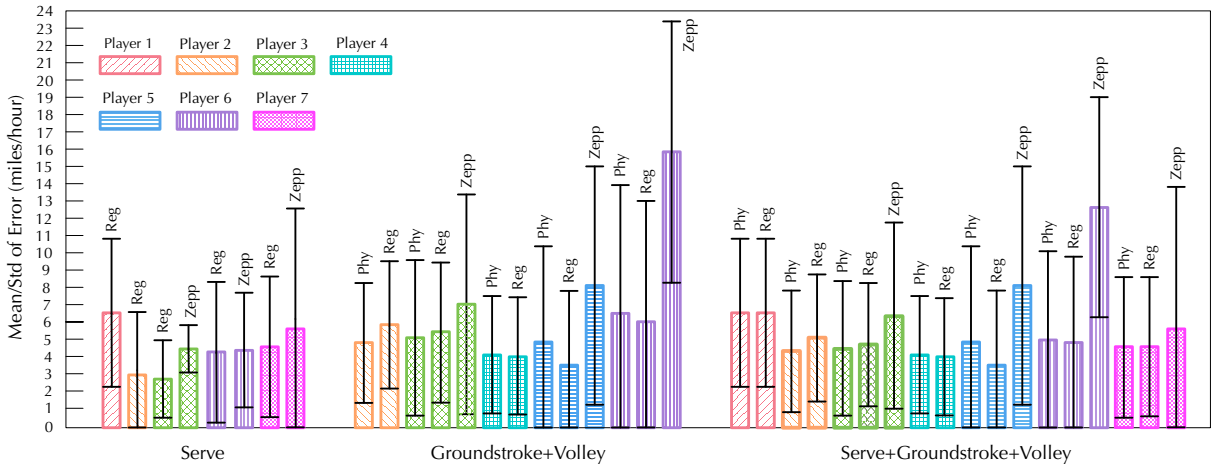

Figure 10: Performance of ball speed estimation models under leave-one-subject-out cross validation

the error of the proposed regression model is 4.3±4.0 miles/hour. The accuracy is 93.4%. For the groundstroke and volley ball speed estimation, the error of the proposed physical model and regression model are 5.0±4.9 miles/hour and 4.9±4.4 miles/hour respectively. The accuracy are 88.4% and 88.8% respectively. For all the tennis shot data, the error of the proposed physical model and regression model are 4.8±4.7 miles/hour and 4.7±4.3 miles/hour, respectively. The accuracy are 90.0% and 90.2%, respectively. From Fig. 10, we find that the performance of the physical model is similar to that of the regression model. For the casual players, the regression model is slightly better. For the regular players, two models have similar performances. For the coaches, the physical model performs slightly better than the regression model. Therefore, we use the regression model to estimate the ball speed for beginner players and the physical model to estimate the ball speed for advanced players.

We compare our ball speed models with two works: the ball speed model in Zepp and a table tennis ball speed model [3]. There are two reasons to choose this table tennis ball speed model for comparison. (1) Both tennis and table tennis belong to racket sports. Compared with other racket sports, table tennis is most similar to tennis. (2) As far as we know, this table tennis model is the only motion sensors-based ball speed model for racket sports. We evaluate the performance of these three models using leave-one-subject-out (LOSO) cross validation. The dataset used for evaluation is collected with both a UG and Zepp sensor, as shown in Table 3. The evaluation results are shown in Table 4. From the table, we find that the proposed physical model and regression model have similar performance. Both of them perform much better than Zepp and the table tennis model. For all the tennis shot data, the proposed physical model is 7.8% more accurate than the table tennis model and 11.7% more accurate than Zepp. The proposed regression model is 8.2%

Table 4: Comparison with State-of-the-art Ball Speed Models

Ball Speed Model	Evaluation Metric	Serve	Groundstroke + Volley	Serve + Groundstroke + Volley
Peter et al. [3]	Mean±Std	11.6±7.1 miles/hour	7.6±6.9 miles/hour	9.1±7.3 miles/hour
	Accuracy	81.4%	81.2%	81.3%
Zepp	Mean±Std	5.2±5.7 miles/hour	10.6±miles/hour	8.9±7.8 miles/hour
	Accuracy	91.8%	70.8%	77.4%
Phy	Mean±Std	N/A	5.3±5.7 miles/hour	4.9±5.1 miles/hour
	Accuracy	N/A	86.7%	89.1%
Reg	Mean±Std	4.3±3.9 miles/hour	5.1±4.9 miles/hour	4.8±4.6 miles/hour
	Accuracy	93.1%	87.4%	89.5%

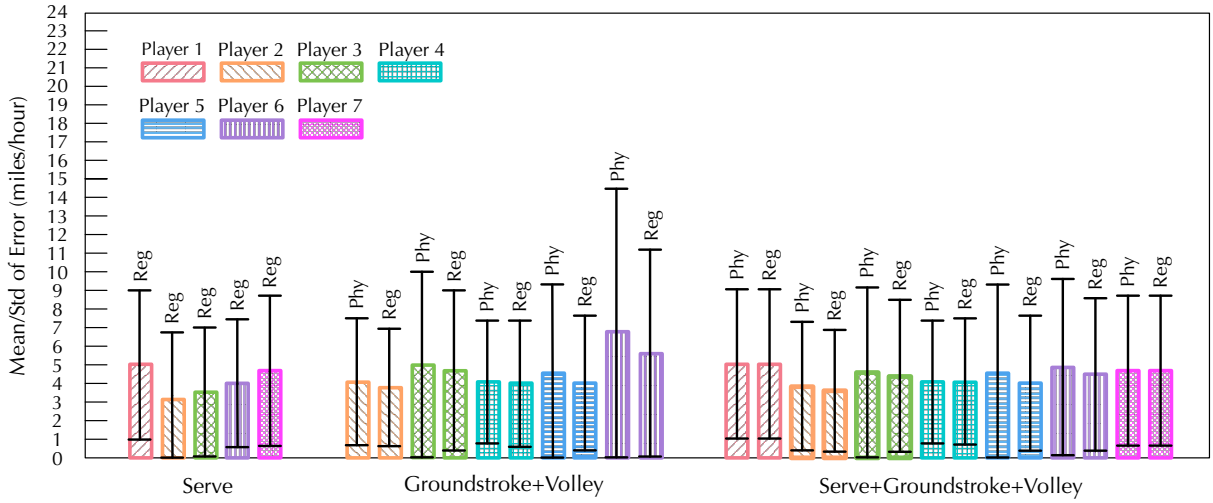

Figure 11: Performance of ball speed estimation models under self test

more accurate than the table tennis model and 12.1% more accurate than Zepp. The table tennis model performs better than Zepp, especially in groundstroke and volley speed estimation. However, it is still outperformed by our models. We think there are two main reasons. (1) Table tennis and tennis are two racket sports that differ a lot in terms of the size, mass, and material of the ball and racket. In addition, the techniques of swinging the rackets are different. In table tennis, a player swings the racket with a lot of wrist action. However, in tennis, a player swings the racket with a lot of body rotation and less wrist action. Therefore, the table tennis model may not be appropriate for tennis. (2) In this table tennis model, the racket speed is modeled as the sum of the wrist linear speed and wrist rotation speed. However, in tennis, the racket speed mainly depends on the rotation of the arm and body. Therefore, our models are more accurate than the table tennis model in characterizing the tennis racket speed.

3.2.2 Self Evaluation. The self test partitions the dataset for each player into 5 randomly chosen subsets of equal size. Four subsets are used to train the model, and the remaining one is used to validate the model. This process is repeated 5 times, such that each subset is used exactly once for validation. The results of the self test are shown in Fig. 11. Mean and standard deviation of error are considered as the evaluation metrics. For the serve ball speed estimation, the error of the proposed regression model is 4.0±3.6 miles/hour. For the groundstroke and volley speed estimation, the error of the proposed physical model and regression model are 4.6±4.6 miles/hour and 4.2±3.9 miles/hour. For all the tennis shot data, the error of the proposed physical model and regression model are 4.4±4.3 miles/hour and 4.2±3.8 miles/hour. Under the self test, both the regression model and the physical model perform better than that under leave-one-subject-out cross validation. In addition, under the self test, the regression model always performs slightly better than the physical model for all the players.

3.2.3 Cross Evaluation. The 5-fold cross-validation test uses all the data to form the dataset. It partitions the dataset into 5 randomly chosen subsets of equal size. Four subsets are used to train the model, and the remaining one is used to validate the model. This process is repeated 5 times such that each subset is used exactly once for validation. The results of the 5-fold cross validation are shown in Fig. 12. Mean and standard deviation of error are considered as the evaluation metrics. For the serve ball speed estimation, the error of the proposed regression model is 4.1±3.8 miles/hour. For the groundstroke and volley speed estimation, the error of the proposed physical model and regression model are 4.8±4.8 miles/hour and 4.6±4.0 miles/hour. For all the tennis shot data, the error of the proposed physical model and regression model are 4.6±4.5 miles/hour

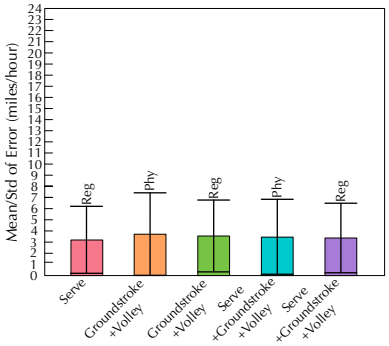

Figure 12: Performance of ball speed estimation models under 5-fold cross validation

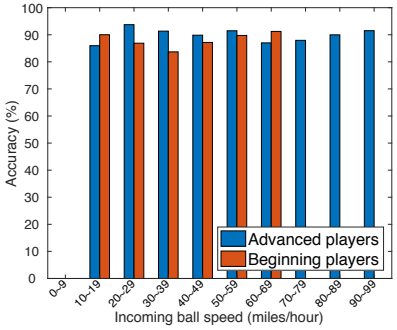

Figure 13: Performance of the physical model under different incoming ball speeds

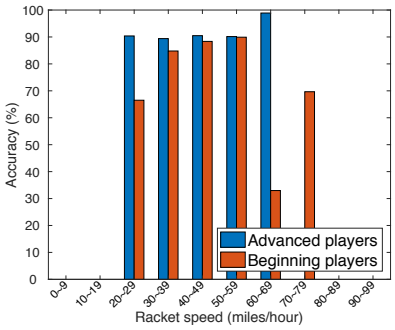

Figure 14: Performance of the physical model under different racket speeds

and 4.4±3.9 miles/hour. Under the 5-fold cross-validation test, both the regression model and the physical model perform better than that under leave-one-subject-out cross validation, and worse than that under self test. In addition, under the 5-fold cross-validation test, the regression model always performs slightly better than the physical model.

3.2.4 Influence Factors on the Physical Model. We evaluate the performance of the physical model under different incoming ball speeds and racket speeds using the leave-one-subject-out cross validation. The results are shown in Fig. 13 and Fig. 14. From Fig. 13, we find that the physical model performs well for both advanced and beginner players under various incoming ball speeds. Therefore, the incoming ball speed factor does not have significant influence on the accuracy of the physical model. From Fig. 14, we find that the physical model performs well for advanced players under various racket speeds. However, when the racket speed is too low (< 30 miles/hour) or too high (≥ 60 miles/hour), the physical model does not perform well for beginner players. The reduction of the accuracy results from the awkward and incorrect swing gestures performed by beginner players when they swing the racket too slow or too fast.

3.3 Stroke Detection Accuracy

Fig. 15 shows the performance of the stroke detection under different thresholds. Three evaluation metrics are considered: precision, recall, and F-measure. F-measure considers the balance between precision and recall as described in Eq. 12:

$$F\text{-}measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

From the figure, we find that as the threshold increases, the precision increases and the recall decreases. When the threshold is $9g$, the F-measure is at highest value: 97.8%. Therefore, we choose $9g$ as the threshold for stroke detection. Under this threshold, the precision and recall are 98.5% and 97.2%. The promising results show the effectiveness of the proposed stroke detection method.

3.4 Stroke Classification Accuracy

We apply the WEKA machine-learning suite [11] to train five commonly used classifiers. The classifiers include AdaBoost (run for 100 iterations), Naive Bayes, SVM (with polynomial kernels), Decision Tree (equivalent to C4.5 [27]), and Random Forests (100 trees, 4 random features each) [42]. Table 5 shows the performance of these five classifiers under 5-fold cross validation test. Same as stroke detection, precision, recall, and F-measure are considered as the evaluation metrics. From the table, we find that the Random Forest classifier performs the best. The precision, recall, and F-measure are 96.2%, 98.2%, and 97.2%, respectively. Therefore, we choose the Random Forest classifier for stroke classification.

Table 5: Comparison of Machine Learning Algorithms for Stroke Classification

Algorithm	Precision	Recall	F-Measure
AdaBoost	92.1%	92.5%	92.3%
Naive Bayes	79.6%	71.6%	73.9%
SVM	92.1%	94.2%	92.9%
Decision Tree	83.4%	85.3%	84.4%
RandomForest	**96.2%**	**98.2%**	**97.2%**

3.5 TennisEye Performance

Fig. 16 shows the overall performance of TennisEye system and Zepp using the leave-one-subject-out cross validation. Mean and standard deviation of error are considered as the evaluation metrics. The cumulative distribution functions of TennisEye and Zepp are shown in Fig. 17. The error of TennisEye is 5.6±4.9 miles/hour, while that of Zepp is 8.9±7.8 miles/hour. The accuracy of TennisEye is 88.2%, while that of Zepp is 77.4%. Therefore, TennisEye is 10.8% more accurate than Zepp in terms of the accuracy.

4 DISCUSSION AND FUTURE WORK

In our physical model, we make several assumptions. For example, we assume that the tennis ball horizontally hits and bounces off the racket. In addition, we assume that the ball impacts at the center of the racket face. We make these two assumptions based on the observations that coaches always swing the racket horizontally and hit the tennis ball at the sweet spot of the racket. However, we also observe that the casual players often swing the racket at different angles and hit the tennis ball at bad positions, i.e., racket frame.

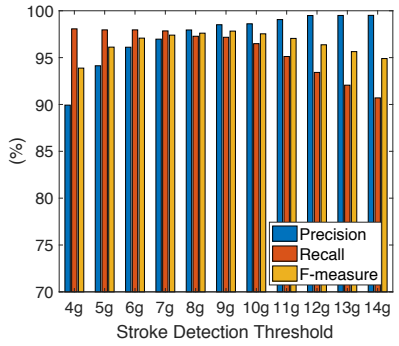

Figure 15: Performance of stroke detection under different thresholds

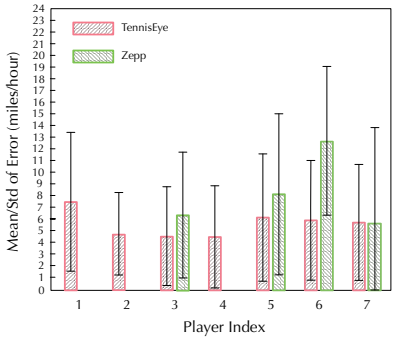

Figure 16: Performance of TennisEye

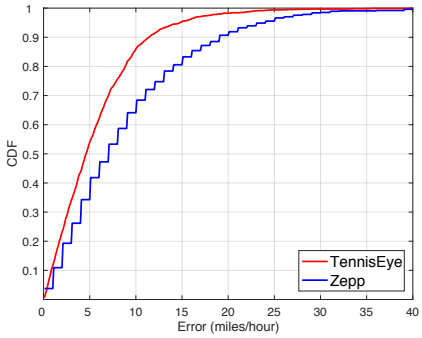

Figure 17: Cumulative distribution functions of TennisEye and Zepp

For casual players, the large variance of their tennis swings and bad hit positions significantly reduce the accuracy of the proposed physical model. Thus, the physical model does not perform well for casual players. We plan to further improve the accuracy of the physical model by releasing these two assumptions.

In our study, we only consider the horizontal impact between a tennis racket and a tennis ball. However, the racket not only moves horizontally, but also vertically. During impact, the vertical momentum of the racket converts to the vertical speed and spin speed of the tennis ball. It would be interesting to explore how much momentum are converted to the vertical speed and how much momentum are converted to the spin speed. We plan to take the vertical speed of the racket into consideration to further improve the accuracy of the ball speed estimation model. In addition, we also plan to estimate ball spin speed in the future.

In addition, the impact process between a tennis ball and a tennis racket is very short. To accurately measure the racket vibration during the impact, a high sampling rate of the UG sensor is needed. The sampling rate in our system is set to be 100Hz. It would be interesting to investigate whether the accuracy will be further improved or not with a higher sampling rate. This will be explored in the future.

5 RELATED WORK

There are mainly two ways to analyze tennis shots. One way is to use computer vision technology. The other way is to use wearable motion sensors.

In computer vision technology, some researchers use one camera to study the trajectory of tennis ball. For example, Yan et al. introduce a methodology to process low quality single camera videos and track a tennis ball with the aid of a modified particle filter [39]. Qazi et al. combine machine learning algorithms with computer vision techniques to predict ball trajectories [26]. Wang et al. employ a neural network approach to predict ball trajectories [33]. These one camera-based methods can only capture video data in two dimensions. The lack of the video data in the third dimension makes data not usable for high precision applications such as line calling or player performance analysis. For this reason, some researchers use multiple dimensional video data to study the tennis shots. Pingali et al. are pioneers to build a multiple camera real time ball tracking system based on six cameras [23]. So far, there have been several commercial products for three dimensional tennis ball tracking and line calling, such as Hawk-eye technology [8] [21] and PlaySight [25]. Hawk-eye technology achieves extremely precise results (mean error rating of 2.6 mm) as it employs 6 to 10 high-dimensional cameras to equip the court. This technology has been applied by many researchers [34] [35] [36]. In addition to Hawk-eye, PlaySight is another multi-camera based sports video-review and analytics system [25]. This system is equipped with six high-dimensional cameras, and uses advanced image processing and analytical algorithms to capture and log stroke type, ball trajectory, speed and spin, in-depth shot data, player movement and more. The computer vision-based technology requires that a tennis court is equipped with camera systems. However, most tennis courts are not equipped with such systems. Therefore, most players can not get access to these systems, which limits the popularity of this technology.

Recent trends show that inertial motion sensors are being used to analyze tennis shots. Some works focus on the stroke behaviors classification. They classify tennis stroke types into forehand, backhand, serve, volley, smash, top spin, and back spin [38] [22] [5] [17] [2]. A variety of machine learning algorithms have been applied, including SVM [38], longest common subsequence [5], Random Forest [17], CNN, and Bidirectional Long Short-Term Memory Networks (BLSTM) [2]. In addition to stroke classification, some researchers assess the performance of tennis strokes and provide recommendations to improve tennis skills [31][40][28]. Srivastava et al. study the consistency of tennis shots [31]. The authors provide recommendations on wrist rotation based on shots performed by professional players. TennisMaster used Hidden Markov Model to segment a tennis serve into 8 phases [40]. By studying the power, gesture, and rhythm for each phase, the authors build a regression model which outputs the score of a serve. Sharma et al. segment a tennis serve into various phases, including backswing, pronation, and follow-through [28]. By comparing the serve phases from a user and that from professionals, the system provide the user with corrective feedback and insights into their playing styles. Different from stroke behaviors classification and performance assessment, we investigate the interaction between the racket and the ball. To our knowledge, none of the previous works use motion sensors to estimate tennis ball speed.

In addition to these research works, there are also some commercial products on the market that assess the performance of the players, such as Zepp [41], Usense [32], Babolat Play [24], and Babolat Pure Drive [7]. These products either integrate the motion sensors inside the racket [7], or require users to attach the motion sensors onto the racket [41] [32] [24]. They analyze the tennis data and compute the key performance metrics for each swing, such as stroke type, ball speed, ball spin, and sweet spot. However, none of these commercial products opens their algorithms to the public. In addition, we compare our proposed ball speed calculation algorithm with the ball speed calculation algorithm in Zepp. The evaluation results show that our algorithm is more accurate than that in Zepp.

6 CONCLUSION

In this paper, we propose TennisEye, a tennis ball speed calculation system using a racket-mounted sensor. It detects tennis strokes, recognizes stroke types, and calculates the ball speed. We propose a regression model to estimate the serve speed. In addition, we propose two models, a regression model and a physical model, to estimate the groundstroke and volley ball speed for beginner and advanced players, respectively. For the leave-one-subject-out cross-validation test, experiments with human subjects show that the TennisEye is 10.8% more accurate than the state-of-the-art work. TennisEye is promising and has commercial potential as it is lightweight and more accurate than the existing commercial product.

REFERENCES

[1] Amin Ahmadi, Edmond Mitchell, Francois Destelle, Marc Gowing, Noel E O'Connor, Chris Richter, and Kieran Moran. 2014. Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors. In *Proceedings of the IEEE BSN*. IEEE, 98–103.

[2] Akash Anand, Manish Sharma, Rupika Srivastava, Lakshmi Kaligounder, and Divya Prakash. 2017. Wearable Motion Sensor Based Analysis of Swing Sports. In *Proceedings of the IEEE ICMLA*. IEEE, 261–267.

[3] Peter Blank, Benjamin H Groh, and Bjoern M Eskofier. 2017. Ball speed and spin estimation in table tennis using a racket-mounted inertial sensor. In *Proceedings of the ACM ISWC*. ACM, 2–9.

[4] Howard Brody. 1997. The physics of tennis. III. The ball–racket interaction. *American Journal of Physics* 65, 10 (1997), 981–987.

[5] Lars Büthe, Ulf Blanke, Haralds Capkevics, and Gerhard Tröster. 2016. A wearable sensing system for timing analysis in tennis. In *Proceedings of the IEEE BSN*. IEEE, 43–48.

[6] Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor. 1978. *A practical guide to splines*. Vol. 27. Springer.

[7] Babolat Pure Drive. 2019. https://www.babolat.us/product/tennis/generic/pure-drive-play-102229.

[8] Hawk eye innovations. 2018. http://www.hawkeyeinnovations.co.uk/sports/tennis.

[9] Biyi Fang, Nicholas D Lane, Mi Zhang, and Fahim Kawsar. 2016. Headscan: A wearable system for radio-based sensing of head and mouth-related activities. In *Proceedings of the ACM/IEEE IPSN*. IEEE, 1–12.

[10] Benjamin H Groh, Frank Warschun, Martin Deininger, Thomas Kautz, Christine Martindale, and Bjoern M Eskofier. 2017. Automated ski velocity and jump length determination in ski jumping based on unobtrusive and wearable sensors. *Proceedings of the ACM IMWUT* 1, 3 (2017), 53.

[11] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *Proceedings of the ACM SIGKDD* 11, 1 (2009), 10–18.

[12] Tian Hao, Guoliang Xing, and Gang Zhou. 2015. RunBuddy: a smartphone system for running rhythm monitoring. In *Proceedings of the ACM Ubicomp*. ACM, 133–144.

[13] Global Wearable Devices in Sports Market. 2018. https://www.researchandmarkets.com/research/k8p2gz/global_wearable?w=4.

[14] Aftab Khan, James Nicholson, and Thomas Plötz. 2017. Activity Recognition for Quality Assessment of Batting Shots in Cricket using a Hierarchical Representation. *Proceedings of the ACM IMWUT* 1, 3 (2017), 62.

[15] Cassim Ladha, Nils Y Hammerla, Patrick Olivier, and Thomas Plötz. 2013. ClimbAX: skill assessment for climbing enthusiasts. In *Proceedings of the ACM Ubicomp*. ACM, 235–244.

[16] Industrial Wearable Devices Market. 2018. https://www.researchandmarkets.com/research/nrtbv9/global_industrial?w=4.

[17] Miha Mlakar and Mitja Luštrek. 2017. Analyzing tennis game through sensor data with machine learning and multi-objective optimization. In *Proceedings of the ACM ISWC*. ACM, 153–156.

[18] Frank Mokaya, Roland Lucas, Hae Young Noh, and Pei Zhang. 2016. Burnout: a wearable system for unobtrusive skeletal muscle fatigue estimation. In *Proceedings of the ACM/IEEE IPSN*. IEEE, 1–12.

[19] Andreas Möller, Luis Roalter, Stefan Diewald, Johannes Scherr, Matthias Kranz, Nils Hammerla, Patrick Olivier, and Thomas Plötz. 2012. Gymskill: A personal trainer for physical exercises. In *Proceedings of the IEEE PerCom*. IEEE, 213–220.

[20] Conservation of Linear Momentum. 2018. https://en.wikipedia.org/wiki/Momentum.

[21] NEIL Owens, C Harris, and C Stennett. 2003. Hawk-eye tennis system. In *Proceedings of the IET VIE*. IET, 182–185.

[22] Weiping Pei, Jun Wang, Xubin Xu, Zhengwei Wu, and Xiaorong Du. 2017. An embedded 6-axis sensor based recognition for tennis stroke. In *Proceedings of the IEEE ICCE*. IEEE, 55–58.

[23] Gopal Pingali, Agata Opalach, and Yves Jean. 2000. Ball tracking and virtual replays for innovative tennis broadcasts. In *Proceedings of the IEEE Pattern Recognition*, Vol. 4. IEEE, 152–156.

[24] Babolat Play. 2019. http://en.babolatplay.com/play.

[25] PlaySight. 2018. http://playsight.com/.

[26] Tayeba Qazi, Prerana Mukherjee, Siddharth Srivastava, Brejesh Lall, and Nathi Ram Chauhan. 2015. Automated ball tracking in tennis videos. In *Proceedings of the IEEE ICIIP*. IEEE, 236–240.

[27] J Ross Quinlan. 2014. *C4. 5: programs for machine learning*. Elsevier.

[28] Manish Sharma, Rupika Srivastava, Akash Anand, Divya Prakash, and Lakshmi Kaligounder. 2017. Wearable motion sensor based phasic analysis of tennis serve for performance feedback. In *Proceedings of the IEEE ICASSP*. IEEE, 5945–5949.

[29] Sony. 2019. https://www.sony.com.au/microsite/tennis/.

[30] Least squares. 2018. https://en.wikipedia.org/wiki/Least_squares.

[31] Rupika Srivastava, Ayush Patwari, Sunil Kumar, Gaurav Mishra, Laksmi Kaligounder, and Purnendu Sinha. 2015. Efficient characterization of tennis shots and game analysis using wearable sensors data. In *Proceedings of the IEEE SENSORS*. IEEE, 1–4.

[32] Usense. 2018. http://www.ubc-tech.com/en/index.html.

[33] Qizhi Wang, KangJie Zhang, and Dengdian Wang. 2014. The trajectory prediction and analysis of spinning ball for a table tennis robot application. In *Proceedings of the IEEE CYBER*. IEEE, 496–501.

[34] Xinyu Wei, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. 2013. Predicting shot locations in tennis using spatiotemporal data. In *Proceedings of the IEEE DICTA*. IEEE, 1–8.

[35] Xinyu Wei, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. 2013. Sweet-spot: Using spatiotemporal data to discover and predict shots in tennis. In *Proceedings of the Annual MIT Sloan Sports Analytics Conference*.

[36] Xinyu Wei, Patrick Lucey, Stephen Vidas, Stuart Morgan, and Sridha Sridharan. 2014. Forecasting events using an augmented hidden conditional random field. In *Proceedings of the Springer ACCV*. Springer, 569–582.

[37] Graham Weir and Peter McGavin. 2008. The coefficient of restitution for the idealized impact of a spherical, nano-scale particle on a rigid plane. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 464. The Royal Society, 1295–1307.

[38] David Whiteside, Olivia Cant, Molly Connolly, and Machar Reid. 2017. Monitoring hitting load in tennis using inertial sensors and machine learning. *International journal of sports physiology and performance* 12, 9 (2017), 1212–1217.

[39] Fei Yan, W Christmas, and Josef Kittler. 2005. A tennis ball tracking algorithm for automatic annotation of tennis match. In *Proceedings of the BMVC*, Vol. 2. 619–628.

[40] Disheng Yang, Jian Tang, Yang Huang, Chao Xu, Jinyang Li, Liang Hu, Guobin Shen, Chieh-Jan Mike Liang, and Hengchang Liu. 2017. TennisMaster: an IMU-based online serve performance evaluation system. In *Proceedings of the ACM AH*. ACM, 17.

[41] Zepp. 2018. http://www.zepp.com/tennis/.

[42] Shibo Zhang, William Stogin, and Nabil Alshurafa. 2018. I sense overeating: Motif-based machine learning framework to detect overeating using wrist-worn sensing. *Information Fusion* 41 (2018), 37–47.

[43] Hongyang Zhao, Shuangquan Wang, Gang Zhou, and Daqing Zhang. 2018. Ultigesture: A Wristband-based Platform for Continuous Gesture Control in Healthcare. *Smart Health* (2018).

[44] Bo Zhou, Harald Koerger, Markus Wirth, Constantin Zwick, Christine Martindale, Heber Cruz, Bjoern Eskofier, and Paul Lukowicz. 2016. Smart soccer shoe: monitoring foot-ball interaction with shoe integrated textile pressure sensor matrix. In *Proceedings of the ACM ISWC*. ACM, 64–71.