Estimating the Speed of Cricket Balls Using Monocular Vision

Abhijay Nair
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623
an1147@rit.edu

I. Introduction

Estimating the depth and distance of objects in videos has been subjected to extensive research due to the rise in popularity of sports, autonomous vehicles and robotics in healthcare. In cricket, the application of perceiving the 3-dimensional trajectory of the cricket ball is enabled by using the Hawk-Eye system [1] that exploits the views from multiple positions to identify the motion and depth information of the cricket ball. Furthermore, knowing the speed of the ball is an important aspect of improving a player's performance. However, this setup is economically infeasible if the game is not sponsored and played officially.

In this paper, we aim to employ a monocular camera to estimate the depth of a bowled cricket ball. This data is ultimately used to determine the speed at which it is thrown. Studies have shown promising results in localizing the position of tennis balls and badminton shuttlecocks in the 2-dimensional image space [2], [3]. This paper aims to further these works by utilizing the pinhole camera model to extrapolate the depth of cricket balls[4]. The described problem is not trivial as a standard cricket ball has a diameter of about 2.8 inches and when bowled, the speed can go up to 160 kilometers per hour (kph). This poses a significant hurdle in estimating the depth for two reasons. The first issue is that since the physical dimensions of the ball are relatively smaller, the accuracy of the localization models is affected [3]. Secondly, as the frames per second of the source video lowers, the possibility of the balls being represented as after images is higher.

Hence, to solve this problem, we use the speed formula to naively compute the baseline speed of the ball. Additionally, a Recurrent Neural Network is trained to regress the speed from the positions of the ball in three dimensions in subsequent frames of the video. Considering the mentioned methods, we can create a baseline speed estimation model using a single camera. The key contributions of this paper are:

- We collected a dataset of bowled cricket balls in 4K resolution at 60 frames per second with 203 samples. Each video has a corresponding ground truth speed of the ball collected using a radar gun. Additionally, the samples have been annotated with bounding boxes signifying the position of the ball before it bounces on the ground.
- The ability of the pinhole camera model is researched to estimate the depth of objects as they move away from the camera.
- Experimental results show promising results of speed estimation without the use of an economically expensive setup that could be used as a baseline for future studies. The naive speed method has a mean error of about 20 kph. Whereas, initial results with limited data could bring the error down to about 11 kph.

II. Background

In this section, we will describe the most prominent part of the experimental analysis of this project. Particularly, the pinhole camera model how the mathematical model is used to estimate the depth of the cricket ball. Furthermore, the Recurrent Neural Network is chosen as the candidate deep learning model to regress the speed based on the movement of the ball which will be introduced here as well.

A. Pinhole Camera Model

The pinhole camera model is a mathematical model that explains the math behind the approximation of the 3-D coordinates of a scene to a 2-dimensional representation of the view of a scene on the image plane. The pinhole camera is not related to the modern day camera but parts of it have been inspired to the cameras that we know and use now.

In this simplest form, a pinhole camera should have an opening which is a single point through which light enters and hits the sensor or the 2-dimensional plane that captures this information that represents the scene. The opening can be considered how aperture works in modern day cameras. Also, pinhole cameras do not use lenses to converge light rays into the sensor but allows light to naturally pass through the pinhole to create a 2-D mapping of the scene. As a result of this simple result, we obtain the most important relation that will be used throughout this project, that is in simple terms, closer an object is to the camera, the larger it will appear to be. This can be mathematically represented as

$$\frac{D}{d} = \frac{Z}{f} \tag{1}$$

where, Z is the estimated distance of the object from the pinhole, d is the size of the object as is perceived by the camera and D is the actual size of the object. So, by using this relation [4], it can be inferred that, as a particular object starts moving away from the camera, the size of that object will decrease in the 2-D plane as well.

B. Recurrent Neural Networks

Deep neural networks have been predominantly used in applications involving time series data [5], [6] to forecast future events or values. The most commonly used model architecture is the Recurrent Neural Network (RNN), that enables the use of an extra dimension of time and propagates the information learned during previous timesteps to future timesteps. Within the context of this project, it is crucial to understand the previous dimensions and positions of the ball in the image plane, and how with time, the values have changed. Using this time-based learning process, the RNN model can essentially create a mapping from the input trajectory of the ball to the speed it achieved.

III. Related Work

This section will be used to describe some key research papers that have been worked on in the past and how they galvanize the motivation to work on this project.

Archana et. al. [2] used computer vision fundamentals such as background subtraction and threshold filtering to track tennis balls. Building upon that idea, TrackNet [3] discuss the use of using a VGG-based convolutional neural network (CNN) to track tiny objects moving at high speeds. They apply the same to tennis videos and also test it on badminton birdies. These works address the first part of this project but the results from these methods are not reliable enough for depth and speed perception.

On the other hand, Gordon et. al. [7] propose the use of a CNN to learn the depths of objects using a monocular camera, but the target objects are usually stationary and relatively larger than a cricket ball like trees, humans or cars. Furthermore, Alphonse et. al. [4] were able to estimate the distance of objects of relatively smaller sizes at a maximum distance of about a meter. But, as the distance increased, the accuracy degraded and as the size of a cricket pitch is about 22 yards or about 20 meters, accurately determining the depth becomes a difficult task.

Finally, Temiz et. al. [8] and Bell et. al. [9] explore the possibility of tracking objects and estimating the velocity of objects. The limitations of these works are similar, they train and test of larger target objects like a car and the speeds they travel at are relatively lower, making it easier to view the objects in each frame as the maximum speed observed was about 11 miles per hour that is way below what we are trying to work with in this paper.

IV. Implementation

Throughout the process of research on this topic, three major milestones have been achieved that is different from other works in this domain. The first is the preparation of a dataset with ground truth speeds of the cricket balls. Next is the naive speed computation of speed using the speed formula and how this method helps as achieve a baseline estimation method that lets us compare other techniques to solve this problem. The last phase is training a deep learning model and see if it can identify a relationship between the estimated trajectory of the ball and predict its speed.

A. Data Processing

We collected about 214 samples of us bowling the ball across a 22 yard long pitch with a set of wickets placed on the other end. The videos were shot on an iPhone 15 pro at 3840x2160 pixel resolution at 60 fps. Additonally, we set up a 128 beam LiDAR scanner and a stereo camera that recorded RGB frames along with a depth map for each frame. Upon the initial data analysis stage, we found that the LiDAR point clouds were unable to capture the fast moving ball spanning across the entire dataset. But, objects like people and the wickets were recorded, indicating that for an accurate 3-dimensional representation, the target objects has to be large enough or has to have a lower velocity. Further, the data from the stereo camera was much worse. The low frame rate caused the data to only include after images of the ball and the depth map was only able to barely distinguish the background wall present about 40 yards away from the camera from the space in front of it.

Hence, we only utilized the RGB video data captured using the iPhone. The videos were segmented to contain only the section where the ball is bowled and reaches the wickets. The segments were exported at 4k resolution sampled at 24 fps. Also, the ground truth speed was recorded for each ball using a radar gun pointed towards the wickets. After cleaning the data and removing videos with faulty readings caused due to throwing the ball to slow or pointing the radar gun away from the ball trajectory, we were left with about 203 trajectory videos with their corresponding speeds.

B. Speed Estimation using Pinhole Model

When the ball is thrown at timestep t_0, it travels towards the batsman, away from the camera and reaches the batsman at timestep t_f, the ball translates in the XY plane on the image plane. Additionally, using the pinhole camera model, the depth Z at each timestep can be estimated as shown in Figure 1. And, the decreasing size of the ball is also recorded through the annotation. Thus the normalized X, Y and the Z coordinates and the monotonically decreasing size of the ball is used to determining the speed of the ball.

Using the camera intrinsics matrix obtained by performing camera calibration using the checkerboard calibration method for RGB cameras [10], we obtain the focal length along the x and the y axis. Ideally, the focal length should be constant along both axes and mechanical errors could cause a negligible deviation in the values. So, we choose the focal length along the x-axis but either of the two values could be used without affecting the final results too much. Additionally, we also note the optical center of the camera which might be translated to a point other than (0, 0) due to manufacturing errors. Finally,

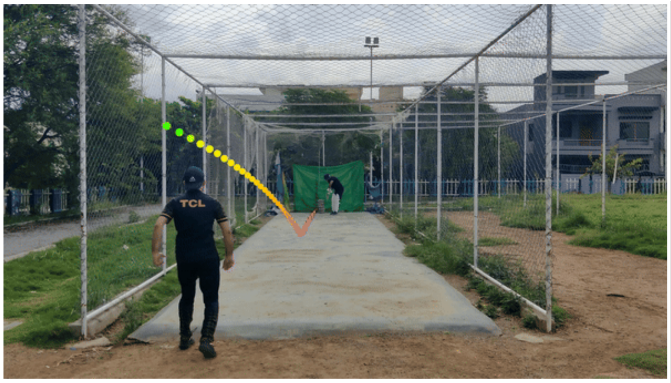

Fig. 1. Depth Estimation using Pinhole Camera Model

we use the optical center and the value of Z obtained using Equation 1 to normalize the X and the Y coordinate position of the ball which can be expressed mathematically as,

$$Z = \frac{f_x * D}{d} \tag{2}$$

$$X, Y = \frac{u - o_x}{Z}, \frac{v - o_y}{Z} \tag{3}$$

Here, u and v are the coordinates of the object center in the image plane. The diameter of the ball is calculated by taking the minimum value of the height and width of the bounding box circumscribing the ball in each frame and can be expressed as

$$d = min(w, h) \tag{4}$$

Finally, we use the values X, Y, Z and d as input variables in the speed estimation experiments. Among these values, we found that using just the change in the diameter of the bounding boxes across frames produced better results than including the positions of the ball as well. But, we formulate the equations in the general form where the required terms could be omitted while implementing.

C. Naive Speed Estimation

Inspired from how high-speed cameras could be used to estimate the speed of objects by tracking them in each frame and dividing the distance traveled by the time taken between each frame, we could obtain the speed of that object. After we obtain this series of speeds using the given trajectory, we compute the median to obtain the estimated speed of each ball trajectory.

Additionally, we tried to compute the mean after removing the outlier values from the speeds using the equation

$$S(x) = \frac{1}{N} \sum_{n=0}^{N} \frac{\|A - B\|}{t_2 - t_1} \tag{5}$$

where $A = (X_2, Y_2, Z_2, D_2)$ and $B = (X_1, Y_1, Z_1, D_1)$ which are the positions and the diameter of the ball at timestep t_2 and t_1 respectively. N is the total number of frames for that ball trajectory and S(x) is the final computed speed for an input trajectory x. If we alter the mentioned equation to compute the median instead of the mean, we obtain the method we used in this experiment.

TABLE I
HYPERPARAMETER TUNING FOR THE RNN ESTIMATION METHOD. EACH MODEL WAS TESTED WITH ADAM AND ADAGRAD OPTIMIZERS

Hyperparameter	**Model A**	**Model B**	**Model C**
RNN_layers	1	2	3
num_neurons	4_24_32_1	4_24_32_16_1	4_24_32_16_8_1

By averaging over the difference between the estimated and actual speed using Equation 6, we compute the mean estimated speed error for the naive speed estimation technique.

$$Err(X) = \frac{1}{N} \sum_{i=0}^{N} |S^i_{est} - S^i_{act}| \tag{6}$$

Where, X represents the entire dataset with N samples with S_{act} ground truth speeds and S_{est} computed speeds. The final estimated error for the entire dataset was estimated to be 12.4 miles per hour.

D. Speed Regression using Deep Learning

The second round of experimentation was performed by training a recurrent neural network in hopes of enabling the model to learn a mapping from a given trajectory to the final speed of that bowling segment. The model was trained using 40 input timesteps and zero padding was used to fill in the positions at the end where enough frames were not available. The model at evaluation stage is able accept trajectories with varying timestep length within 0-40 frames. We performed hyperparameter tuning by changing the number of layers and neurons in each layer of the model along with the optimizer for the model as described in Table 1. The first and the last values in the number of neurons are the input and output neurons and the second value is the number of neurons present in a dense layer before the RNN layers used as a trainable dense embedding for the data. The other intermediate values are the neurons present in each RNN layer. Finally, the Mean Absolute Error between the predicted and actual speed values were used to compute the loss.

Since the entire dataset only had 203 samples, the model was unable to learn a decent mapping from the input to the desired output. The model simply started to output a single value that would optimize the loss over the entire dataset that had a mean error of about 6.8 mph. But, this proves that given an acceptable amount of data, that we are projecting to be about 2000 samples of bowling videos, the model can ultimately learn a speed mapping accurately.

V. DISCUSSION

The two methods described in this paper are not exhaustive and various other techniques can be employed to utilize the available information in estimating the speed of bowled cricket balls. The naive speed estimation method can be seen as a low-overhead version of using a high speed camera whereas the RNN model tries to learn the one-to-one mapping of a trajectory to a predicted speed. Here we will be describing a couple more possible research directions that could be used to estimate the speed of the ball.

- The dataset we used in this paper had a couple of drawbacks such as the lack of enough trajectories and the absence of the cricket pitch markings. One promising improvement that could made is to identify the bounce point of the ball in the real world and also in the video. Once that point is known to us, we can use the actual distance traveled and the time taken for the ball to reach the bounce point to estimate the actual speed of the ball with a higher accuracy than relying on estimated depth values to compute the distance between each frame.
- Physics engines have seen tremendous growth since the past decade, in due regards to the boom of the gaming industry [11]. One way in utilizing the game engines to estimate the speed of a ball would be to emulate the gameplay in the virtual world. Essentially, since it is assumed that the coordinates of the ball is known in the image plane and the availability of the depth information, we can simulate the trajectory of the ball in virtual 3-dimensional world and let the engine take care of computing the velocity of the ball given the discrete positions.
- Finally, we noticed that since the ball is moving too fast, sampling the video at 24 frames per second produced afterimage of the ball in multiple frames, that made the data noisier. To overcome this problem, we propose sampling the video at 60 frames per second or higher to improve the ball localization process that could potentially help in improving in computing the distance of the ball from the camera, thus ameliorating the speed estimation process.

VI. Conclusion

This work paves a way for further research in the domain of estimating the speed of small and fast moving objects by using a monocular camera instead of economically expensive specialized sensors. Furthermore, the experimental analysis described in this paper proves that given a trajectory of the object with its dimensions, it is possible to estimate the depth and compute the speed at which it is traveling. Even though the results are not conclusive to support the adoption of this method for reliable speed estimation, future work in this domain is encouraged to improve the process using the proof of concept explained in this paper.

References

[1] L. Jayalath, "Hawk eye technology used in cricket," *South Asian Research Journal of Engineering and Technology*, vol. 3, no. 2, pp. 55–67, 2021.

[2] M. Archana and M. K. Geetha, "Object detection and tracking based on trajectory in broadcast tennis video," *Procedia Computer Science*, vol. 58, pp. 225–232, 2015.

[3] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng, "Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, pp. 1–8.

[4] P. Alphonse and K. V. Sriharsha, "Depth perception in single rgb camera system using lens aperture and object size: A geometrical approach for depth estimation," *SN Applied Sciences*, vol. 3, no. 6, p. 595, 2021.

[5] A. Nanduri and L. Sherry, "Anomaly detection in aircraft data using recurrent neural networks (rnn)," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*. Ieee, 2016, pp. 5C2–1.

[6] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 2017, pp. 1643–1647.

[7] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8977–8986.

[8] M. Temiz, S. Kulur, and S. Dogan, "Real time speed estimation from monocular video," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39, pp. 427–432, 2012.

[9] D. Bell, W. Xiao, and P. James, "Accurate vehicle speed estimation from monocular camera footage," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 419–426, 2020.

[10] F. Basso, E. Menegatti, and A. Pretto, "Robust intrinsic and extrinsic calibration of rgb-d cameras," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1315–1332, 2018.

[11] A. Chia, "The metaverse, but not the way you think: game engines and automation beyond game development," *Critical Studies in Media Communication*, vol. 39, no. 3, pp. 191–200, 2022.