Cricket Ball Trajectory Estimation for LBW Prediction

Anirudh Bhupalarao

I. Introduction

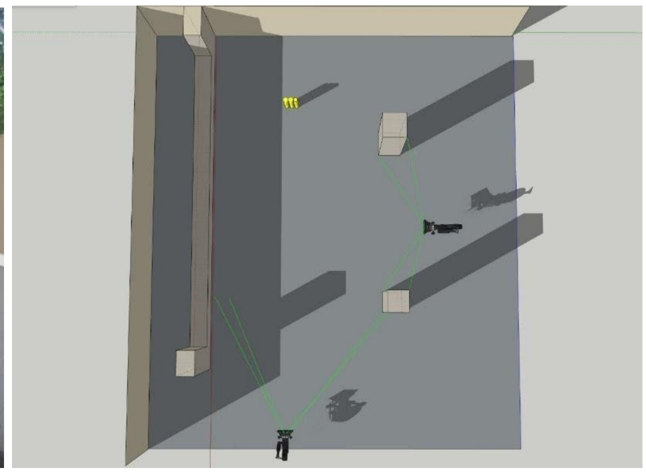
Leg Before Wicket (LBW) is one of the ways in which the batsmen can be dismissed in cricket and has been a subject of controversy historically as there was a death of proper technology. At present, a computer vision technology termed “The Hawk-Eye System” is used to predict the trajectory of the ball and helps in decision making concerning LBWs. Here, we seek to implement an LBW prediction system by predicting the path of the ball towards the stumps from the frames of a video. We first pursue object tracking using simple digital image processing methodologies and then use certain python libraries and fundamental concepts of physics to predict the path of the cricket ball onto the stumps. This methodology has been found to work quite well in our local setting and has given us very favorable results.

II. Objective/Aim

In this project, our aim is to predict the trajectory of the ball and conclude whether an LBW (leg before wicket) has occurred using this predicted path. For this we first have to detect the object (in this case a red cricket ball) and isolate it from the background by some means. Then, we find an appropriate way to characterize the center of the ball for usage in the last step. The last step involves the path prediction algorithm which takes into consideration the array of centers of the ball at each time step and predicts the trajectory for future time steps.

III. Methodology and Results

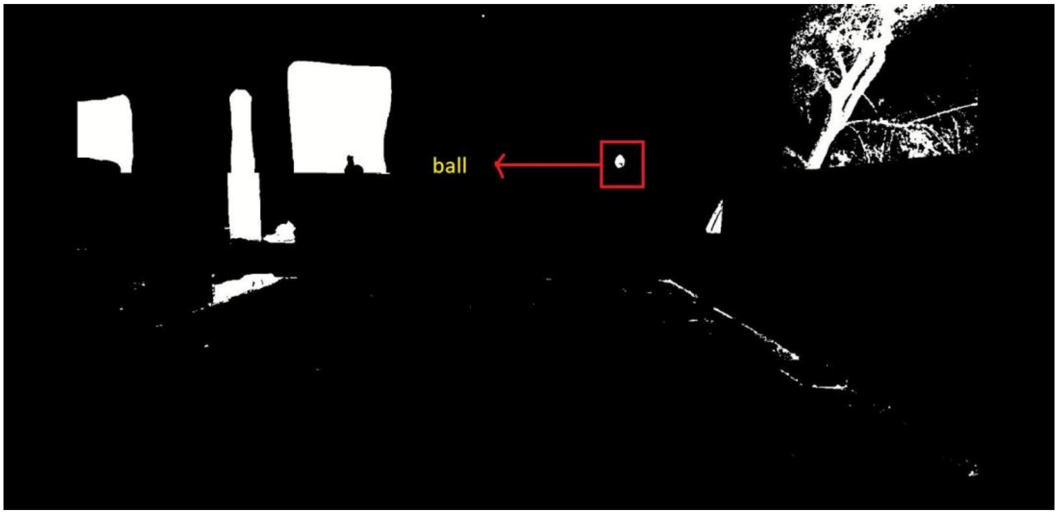
An appropriate database containing around twenty videos of the ball being bowled towards the stumps are taken by us. These are videos taken by us in a local setting keeping in mind the contrast, brightness, frames per second (fps) etc. The implementation can be broadly categorized into three stages. In all these stages we shall present an example of a single frame to be worked upon and see the results side by side along with the methodology. The frame we have taken is given below along with the camera setup:



First Stage

Here, we deal with detection of the ball frame by frame. The steps are as follows:

- Convert the RGB image into grayscale for ease of detection in later steps
- Using the grayscale image we use the binary inverse thresholding function to convert any pixel having an intensity of less than 70 (on a 255 scale) to white (or intensity value of 255) and anything having an intensity of more than 70 to black (or intensity value of 0).
- Since the ball is of dark red color the grayscale image would lend it an intensity value closer to 0 than to 255. So, when we apply thresholding, the black color of the ball changes to white in the output. The relatively lighter color of the background is thresholded to an intensity value of 0 (black)

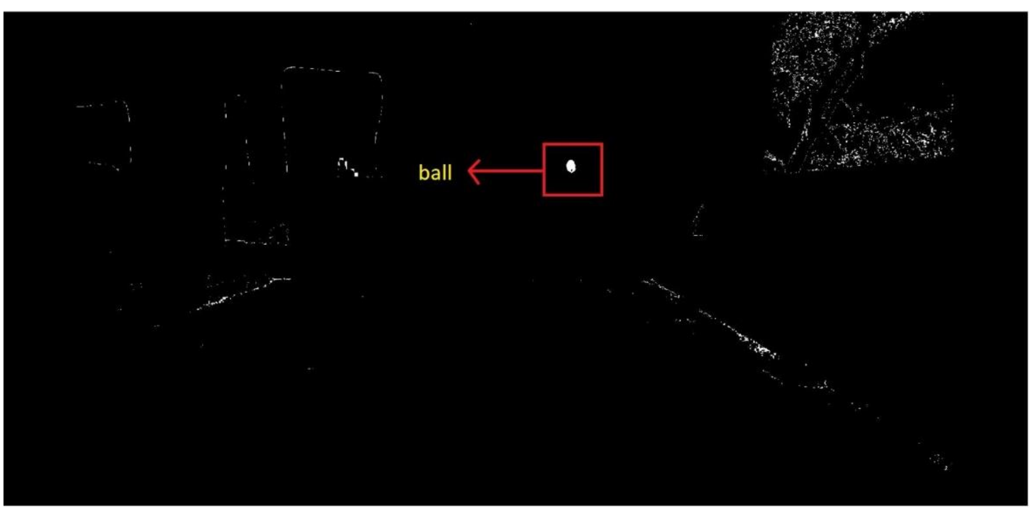


Second stage

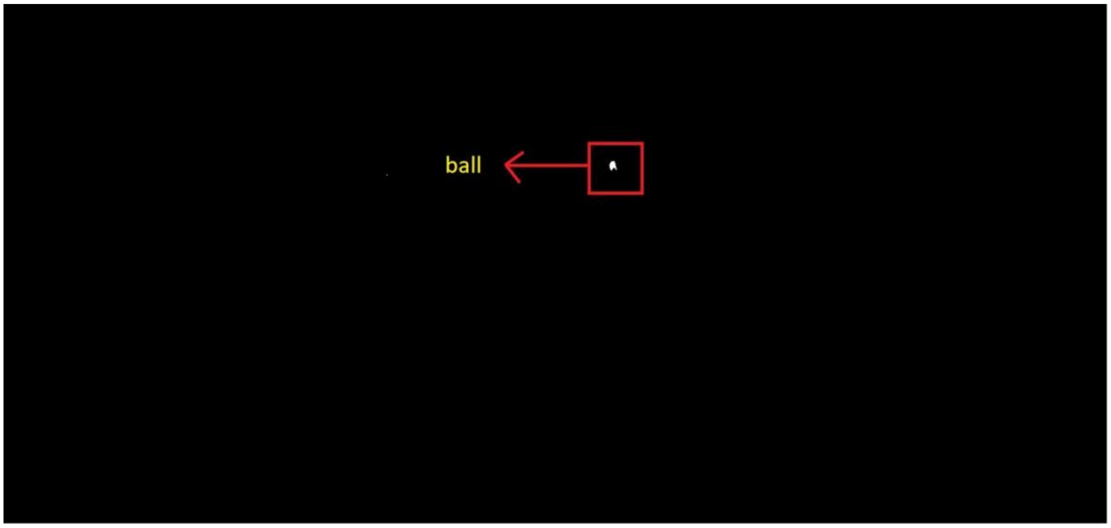
- In the thresholding operation we found that many background elements have also been set to white indicating that we need some kind of motion based elimination method by which the ball is identified while the rest of the background entities are removed
- For this, we employ a pixel-based operation where the pixels from the previous frame are compared to the pixels in the current frame based on the truth table below

f_o	f_i	f_i'
0	0	0
0	1	1
1	0	0
1	1	0

- f0 is the previous frame, fi is the current frame and fi’ is the resultant for whether there is an object in motion or not.
- Therefore, if by this algorithm, the pixel is judged to not be a part of a moving object, then its intensity is set to 0 (black)
- Image after this operation is shown below:



- We still see some small white pixel components coming from the background elements as these may have moved a bit during the previous operation
- We characterize this as salt noise and proceed to remove it by using median filtering technique with a filter size of 5x5
- Then we use OpenCV methods to find contours in the frame. Contours represent a curve joining the continuous points along the boundary, having same color or intensity of pixels
- The result of the previous two steps are shown below:

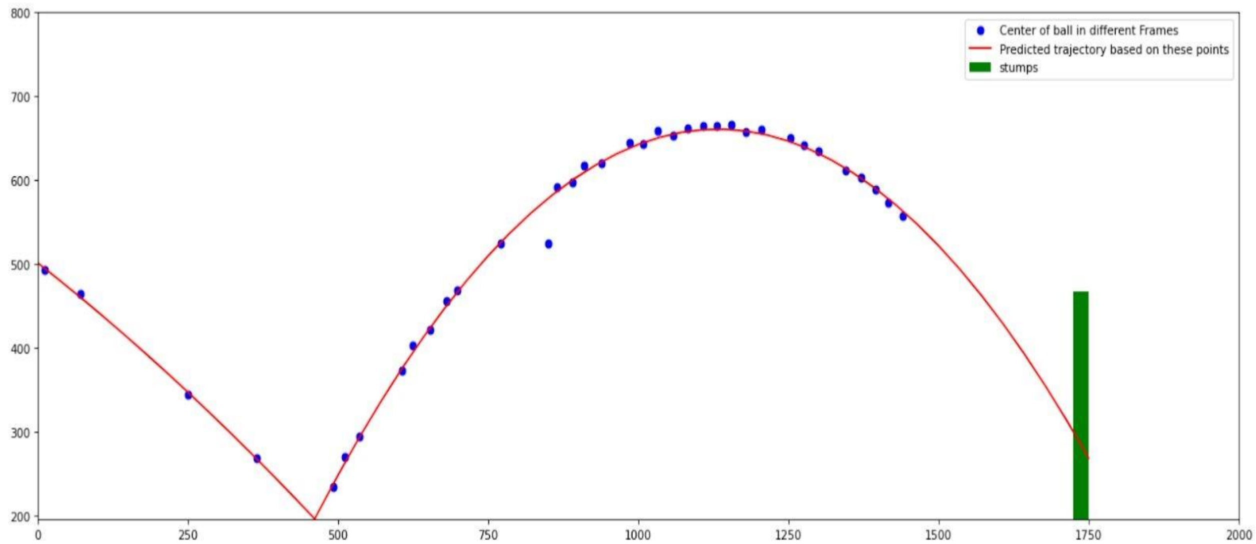


- Some of the edge details of the ball are lost due to the median filtering step but the center part of the ball still remains intact.
- We then need the coordinate information about the ball for use in the third stage. We get a very good approximation of the center of the ball by taking the center of mass of the white pixels as follows:

  x_com = $\frac{\sum(distance\ of\ i^{th}\ white\ pixels\ along\ x)}{total\ white\ pixels}$ y_com = $\frac{\sum(distance\ of\ i^{th}\ white\ pixels\ along\ y)}{total\ white\ pixels}$

Third Stage

- In this stage we hope to predict the path of the ball by approximating it to that of a polynomial of degree two. Newton's equations have hinted that projectile motions such as these can be characterized by second degree polynomial equations, so this approximation is very close to the true path.
- We have used python libraries such as NumPy and matplotlib to plot the positions of the ball at each frame and then predict the trajectory.
- The program takes in the listen of coordinates of the center of masses as calculated in the previous stage for each frame. A second degree polynomial curve is made to fit these points by optimizing for minimum least square error and thus gives us the result as shown below (The labels for the different parts of the graph are given on the top-right):



- We can see that even after the blue dots, the path is predicted.

IV. Discussion and Conclusion

We have done the above procedure with 20 different videos and have achieved favorable results. The techniques implemented in the project involve basic digital image processing methods learnt during the lectures. Some ways in which this method may not work:

- There may be some outliers or noisy points when we start detecting the centers of the ball in each frame and this may cause minor disturbances
- A camera with high frames per second (fps) capturing capability is preferred as we would want to capture the fast-moving ball. (In the above results a phone camera was used. Nevertheless, we achieved favorable results)
- Dark moving objects in the background may be mistaken to be the ball and the results will not pan out as expected

In conclusion, we have built a program that does well as long as some conditions are taken care of. More advanced works have been carried out in the field which give very good results and are being used in cricket matches today. This method serves as an example that uses only the very fundamentals of digital image processing and yet achieves very good results.

V. References

[1] M.A. Zaveri, S.N. Merchant, U.B. Desai, Small and fast-moving object detection and tracking in sports video sequences, 2004 IEEE International Conference on Multimedia and Expo, Taipei, Taiwan.

[2] X Yu, C Xu, H.W. Leong, Q Tian, Q Tang, K.W. Wan, Trajectory-Based ball detection and tracking with applications to semantic analysis of broadcast soccer video, 11th ACM international conference on Multimedia, 2003.

[3] X Yu, Q Tian, K.W. Wan, A novel ball detection framework for real soccer video, ICME 2003.

[4] G. Kuhne, S. Richter, and M. Berer. Motion-based segmentation and contour-based classification of video objects, In Proc. ACM MM, Canada, 2001.

[5] N Singla, Motion Detection based on frame difference method, International Journal of Information and Computation Technology, 2014.

[6] Z Zhang, De Xu, Min Tan, Visual measurement and prediction of ball trajectory for table tennis robot, IEEE Transactions on Instrumentation and Measurement, 2010.

[7] Rafael C. Gonzalez and Richard E. Woods. 2006. Digital Image Processing (3rd Edition). Prentice-Hall, Inc., USA.