

Use of Image Processing Techniques for Making Run out Decision in Cricket

Shashank Yeole^{#1}, Nisheet Sharma^{#2}, Yogesh Shinde^{#3}, Sufiyan Shaikh^{#4}

[#]Department of Computer Engineering
Sinhgad Institute of Technology, Lonavala, India

¹shashank.yeole@live.in

²nisheet.sharma@live.in

³yshinde@gmail.com

⁴sufiyan.shaikh33@gmail.com

Abstract:*In the world renowned game of Cricket, currently for giving umpiring decisions like Stumping and Run Out, the Third Umpire has to review the various angular video footages thoroughly and laboriously. According to the current Laws of Cricket, the Third Umpire is expected to give the correct decision within 30 seconds, although it's in his discretion to take as much time required by him. This process evidently takes a lot more time than 30 seconds. Hence, a software system for assisting the Third Umpire, is required that will help cut short the above time lag. Then, the Third Umpire will just provide the Video feeds as input to the system and it will compute a result, which will aid the Third Umpire in reaching a decision quickly. Our Idea is to use Image Processing techniques which can provide us with the means of building such a software system that can be used for the above mentioned scenario. The Software system will make use of Object tracking algorithm for tracking Wickets, and depending on the displacement of wickets the frames of that moment will be taken using a Frame Extraction algorithm.*

Keywords:*Frame Extraction, Object Tracking, Stumping, Run Out, Third Umpire, Video Feeds.*

I. INTRODUCTION

In the current scenario, whenever the Onfield Umpires are unsure of the decision that is they are in a state of doubt, they ask the Third Umpire to make the decision. The Third Umpire then views the video feeds and analyses them. He has to continuously replay the events in order and watch intently to decide whether the player is out or not out. Normally, the video feed of the default angle is enough to make the decision but in some peculiar cases, he has to redo the same process for a video feed from another angle. This process is very tedious and time consuming. Some of such peculiar cases are: Bat in the air while in a close run-out situation (for this the Umpire has to watch the video from two different cameras and take a decision.), in some close situations the exact moment the bails went off is missing.

To tackle the latter problem, high speed cameras can be put into force. But the main issue with using high speed cameras is that, they are very expensive and so are not accepted by the ICC, but the broadcasters use it for entertainment purposes only.



Fig. 1: Third Umpire

Another system which is used in International Matches by many countries is the D.R.S. The D.R.S. gives the individual teams to challenge the decisions given by Umpires. D.R.S. makes use of Hawk-Eye, Hotspot technologies for making a decision. Mostly in case of LBW decisions, hawk-eye is the technology used. Three things are checked the Pitch of the Ball, Impact of the ball, and Anticipation of Ball hitting/missing the Wickets. All three collectively are used to deduce the result out/not-out/on-field call. For checking whether a player has edged the ball or not the Hotspot technology is used, which highlights the portion of bat the ball has hit, if it actually did hit the bat. Else, the ball has missed the bat completely.

Recently a new technology has emerged and been made part of the International Cricket matches, which is the Zing Wicket System. It was developed in Australia and was first used in Australia's Big Bash League. It was then accepted by the ICC and was used in the recently concluded ICC World T20 Cup 2014. The Zing Wicket system has ingrained LED lights in the bails and the stumps of the wicket which go off whenever the wickets are disturbed. This technology helps the umpires in judging the exact time of impact of the ball in a close run out or stumping situation.

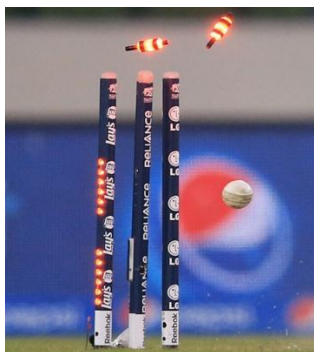


Fig. 2: Zing Wicket System

II. PROBLEM STATEMENT

According to the Laws of Cricket, the Third Umpires have a target time of 30 seconds to come to a decision, while in reality it takes them 50 seconds to more than a minute at times. This adversely affects the rhythm of the game, and this poses as a threat to the time constraints on the game. We strive to cut short the time taken to give the decision by introducing our software system that relies on image processing techniques. Currently we are providing decisions for Stumping and Run out only.

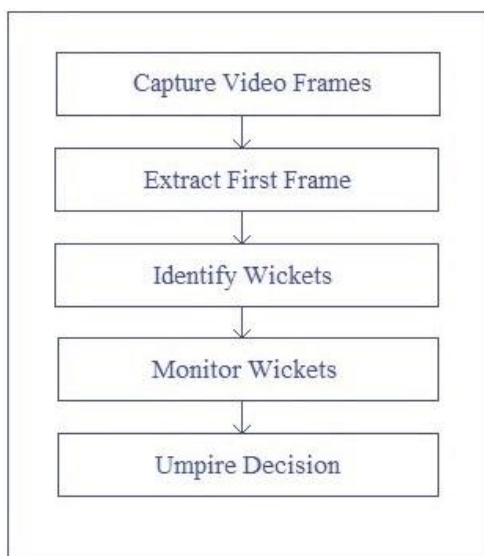


Fig. 3: System Architecture

III. SYSTEM ARCHITECTURE

We are using MATLAB here, to implement the business code. The GUI for the Application is also developed in MATLAB using GUIDE. We will be dealing with two

interfaces namely, Initializing Interface and Monitoring Interface.

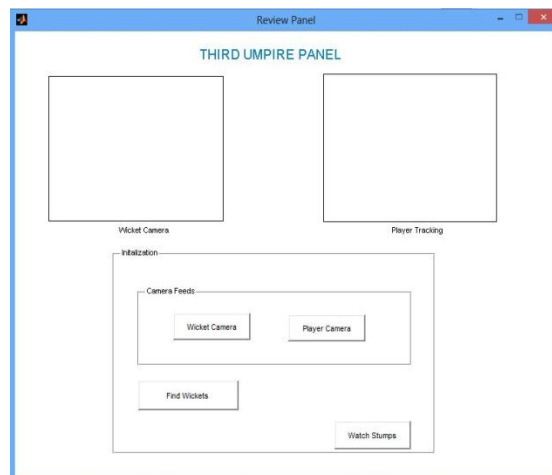


Fig. 4: Pre Initializing Interface

In the Initializing Interface, first the user will provide the Camera feed as input to the application. The feeds from all the cameras, covering different viewing angles is to be taken and processed. After the input is received the Application will identify the wickets, i.e. the pixels of the wickets in the feed will be located. When the application has successfully identified the wickets, the user will receive a notification on the screen. And, the user can click on Watch Stump button on the GUI Screen.



Fig. 5: Post Initializing Interface

The Monitoring Interface succeeds the Initializing Phase. In this, all the dedicated cameras i.e. all the cameras covering the important parts of the situation under consideration, are focused on the stumps. We call this the Watch Stump part. Here, we want to adjudge when there is a displacement of the object under observation. Once we have accurately identified the frame in which there was a displacement, we use that frame number to extract frames from other cameras, to capture the event from different angles.

When the frame is identified, that frame and all other camera frame of that time can be used for giving the decision. The frame can also be checked by the user whether the system has given correct results or not. And with the other camera feeds the user will check whether the player is inside or outside

the crease. Based on which the user will make the decision whether he is out or not and will press the radio button for the respective decision. This system takes out the part of identifying the time at which the bails went off or the wickets were off the ground and saves about 10 seconds for the review panel to make the decision.



Fig. 6: Monitoring Interface

i. Capture Video Frames

In the first step, the video feeds for the desired time period is collected. The desired time period would be the one in which the actual event (stumping or run out) occurred. In our implementation we have made use of RAW avi format videos, which are provided offline. Hence, the frame synchronization of all the cameras was not possible. But, that problem will be resolved if instead of offline feeds, live feeds are provided. The Cameras used are of 12 MP, and they capture video feeds at 30 fps.

ii. Extract First Frame

This is the pre-step for Identifying wickets. From the captured video frames, we select the first frame in the list, for identification of wickets. This can be accomplished by using Frame Extraction. Frame Extraction is the technique of converting a video feed (in our implementation the avi format video file) into individual frames and extracting a certain frame or set of frames from these.

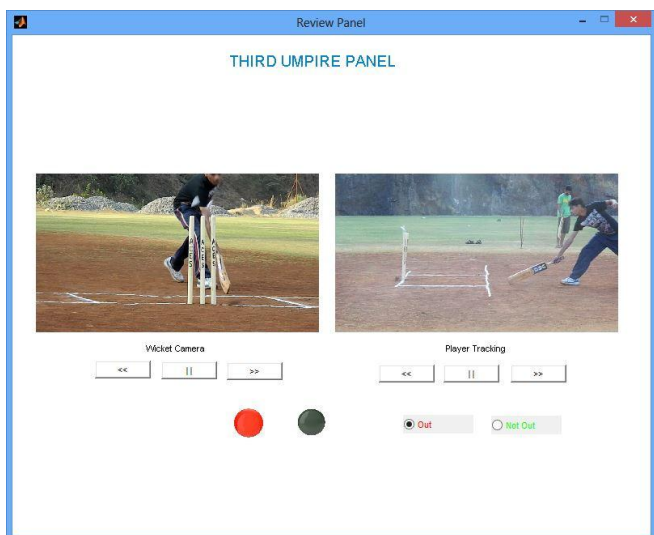


Fig. 7: Decision Interface

This operation will be repeatedly used for Umpiring decision, as we will require multiple frames for comparison, thus we'll need frame extraction at each stage.



Fig. 8: Raw Video Feed

iii. Identify Wickets

After the frame has been selected, we differentiate the pixels of wickets from the rest of the frame pixels. This involves use of Thresholding formula given as follows.

$$I(x,y) = \begin{cases} 1 & D(x,y) \geq T \\ 0 & D(x,y) < T \end{cases}$$

Where, T is the threshold limit.

We then calculate the PPI (Pixels per Inch). In our computation we have manually considered the measurements; we assume the wicket's measurements are near about the following values. Width: 17 pixels, Height: 397 pixels. The Distance between the wickets is assumed to be around, 32 pixels.

Based on the above mentioned values, we try to identify three objects of 17x397 each and placed at a gap of 32 pixels. Please note the above values are near about, and not absolute. These objects are then finalized as the wickets by marking the boundary edge around them as shown in fig. The red lines indicates the boundary edges.

Once identified such objects we store the coordinates of these objects and monitor them for any changes.

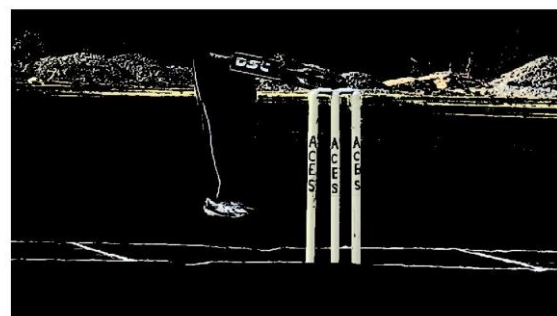


Fig. 9: Frame after Thresholding

iv. Monitor Wickets

To monitor the wickets, we keep track of the wicket pixels, in the remaining frames. As the wicket pixels are stored For this purpose we use Binary Algorithm. In the Binary Algorithm, we take the middle frame of the 30 frames of a second, and compare it with the main frame, if the difference calculated is above the Threshold, its nearby frames are selected, else we choose another frame from the next half and repeat the above process.

v. Umpire Decision

The Wicket monitoring process monitors when the bails have come off. The frame, in which the bails have come off, is captured from all the cameras. The Umpire can rewind & forward the frame sequence to check bails & based on that information he will give a decision. If, he is inside the crease line, the decision is Not Out. In case, the player has touched the crease line, or is out of it, the decision is out.

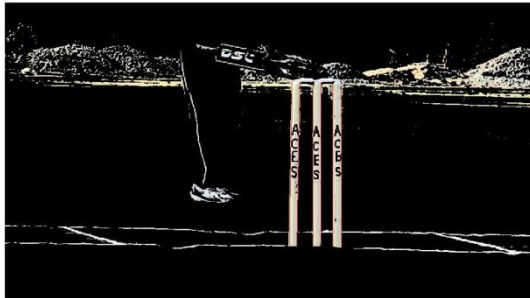


Fig. 10: After Wickets have been identified

IV. RESULTS

For checking the performance of our system we shot 20 test case videos and run those videos through the system as we got the results in less than 4 seconds in all the cases.

Now we compared it with the human eye calculations which took near about 15 seconds to give the result. In each case the system was reducing the time around 88% of which the human eye took to identify the time of the bails off or wickets dislodged. In table 1 we have mentioned some of the test cases and the results of the cases and the reduction in time taken by human eye in those cases.

Table 1: Result

Sr.No	Time by Review Panel	Time by Human Eye	Reduction in Time
1	1.1	12.3	91.06
2	1.3	14.4	90.97
3	1.2	15.1	92.05
4	1.5	13.5	88.89
5	1.4	14.3	90.21

V. PROBLEMS

In our project, we have used offline video feeds, due to which the synchronization of the Cameras was not possible. This could be an issue, but in real time, when live feeds will be available this issue would not exist, as then the feeds will be synchronized frame by frame.

VI. CONCLUSION AND FUTURE WORK

The purpose of the system was to minimize the time taken by the third umpire for providing the decision. This system solves the first part which is identifying the time when the bails went off or the wickets were dislodged.

For the second part of tracking the player at the time of bails went off or wickets dislodged needs can be the future work.

REFERENCES

- [1] H. B. Kim, K. Sim, "A Particular Object Tracking in an Environment of Multiple Moving Objects", International Conference on Control, Automation and Systems 2010, Oct. 27-30, 2010
- [2] T. Mahmood, S. O. Ahmed, M. H. Swaleh, S. H. Nayer, "A-Eye: Automating The Role Of The Third Umpire In The Game Of Cricket", IEEE, 2011
- [3] T. Shepherd, "Cricket Rules: A Player's Guide", Cassell Illustrated, 1995
- [4] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", 2nd Edition, Prentice Hall, January 2002. [Online]. Available: <http://www.worldcat.org/isbn/0201180758>
- [5] B. Janhe, H. Haussecker, "Computer Vision and Applications: A Guide for Students and Practitioners", Academic Press, April 2000
- [6] R. Venkatesan A. Balaji Ganesh, "Real Time Implementation On Moving Object Tracking And Recognition Using Matlab", IEEE, 22-24 Feb. 2012
- [7] S. Prasad, S. Sinha, "Real-time Object Detection and Tracking in an Unknown Environment", IEEE, 2011