"Hawkeye Technology Visualization"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Computer Science and Engineering

Project Mentor:
Mr. Sunil Dhankar
Assistant Professor

Submitted By :
Anjani Kumar Iyer
(17ESKCS022)

Department of Computer Science and Engineering
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota
Session 2020-2021

DECLARATION

We hereby declare that the report of the project entitled Hawkeye Technology Visualization is a record of an original work done by me at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of ”Mr. Sunil Dhankar,(Assistant Professor)” (Dept. of Computer Science and Technology) and coordination of ”Mr. Ankit Kumar”, (Associate Professor)” (Dept.of Computer Science and Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Computer Science and Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Anjani Kumar Iyer
(17ESKCS022)

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mr.Sunil Dhankar. He has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Mr. Ankit Kumar for his co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Prof. Dr. Mukesh Gupta, HOD, Department of Computer Science and Engineering, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Anjani Kumar Iyer
(17ESKCS022)

Contents

List of Figures

Chapter 1

PROJECT INTRODUCTION

1.1 Problem Statement and Objective

Several ball tracking algorithms have been reported in literature. However, most of them use high-quality video and multiple cameras, and the emphasis has been on coordinating the cameras or visualizing the tracking results. This paper aims to develop a system for assisting the umpire in the sport of Cricket in making decisions like detection of no-balls, wide-balls, leg before wicket and bouncers, with the help of a single smartphone camera. It involves the implementation of Computer Vision algorithms for object detection and motion tracking, as well as the integration of machine learning algorithms to optimize the results.

Techniques like Histogram of Gradients (HOG) and Support Vector Machine (SVM) are used for object classification and recognition. Frame subtraction, minimum enclosing circle, and contour detection algorithms are optimized and used for the detection of a cricket ball. These algorithms are applied using the Open Source Python Library - OpenCV. Machine Learning techniques - Linear and Quadratic Regression are used to track and predict the motion of the ball. It also involves the use of open source Python library VPython for the visual representation of the results. The paper describes the design and structure for the approach undertaken in the system for analyzing and visualizing off-air low-quality cricket videos.

1.2 Literature Survey /Market Survey/Investigation and Analysis

Existing System:

- Cricket umpire assistance and ball tracking system that using video from a single smartphone camera.
- Ball Tracking
- Virtualisation

1.3 Introduction to Project

In cricket, an umpire is a person who has the authority to make judgments on the cricket field, according to the laws of cricket. The umpire's decision depends on many criteria, including where the ball pitched, whether the ball hit in line with the wickets and whether the batsman was attempting to hit the ball. The game of cricket is widely played and followed in India and many other countries in the world. There are millions of amateur and aspiring cricketers involved. But the high cost and amount of technological requirements of tracking technologies restricts their use in any matches, competitions and training academies other than the ones operating at an international level. Computer vision seems like a natural choice for these applications. In spite of the success of computer vision technology in several other fields, there are very few computer vision systems and algorithms which enhance the experience of cricket which operate at a low cost.
A promising research direction is the use of computer vision to detect, identify and track the cricket ball (and other relevant objects in the context of cricket), and machine learning techniques to optimize and further predict various results and decisions. The use of just one camera, which may be of a quality equivalent to modern day smartphone cameras, along with the various algorithms and techniques of Computer Vision and Machine Learning can help us achieve a system that reliably assists the umpire and operates at a cheap cost.
This paper aims to develop a low-cost computer system which assists the umpire in cricket, operates at a low cost, has lesser technological (software and hardware) requirements, and can be used at sub-international levels in the sport of cricket.

1.4 Approach

Figure 1 below shows a flowchart of our umpire assistance system. It consists of multiple stages. As a pre-processing stage, image frames are extracted from the video, and the ball is detected in each of the frames. The ball tracking module keeps track of the ball position following which the 3D mapping module maps the tracked ball position from 2D into 3D. This 3D data is then smoothened out using regression techniques. Finally, the visualization module uses this mapped 3D data to determine the umpiring decisions and visualize the results in 3D.

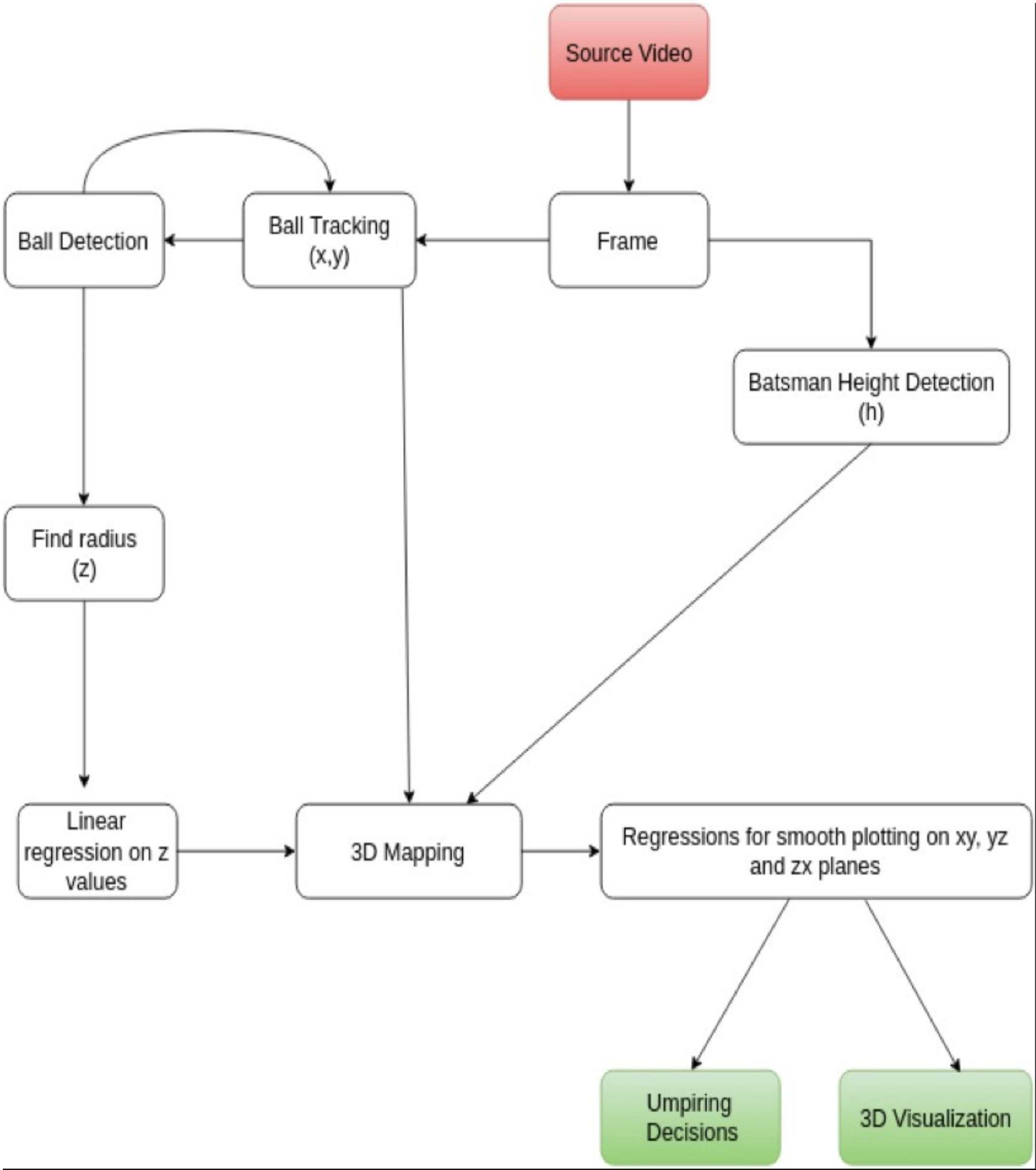

Figure 1.1: Flowchart of the process

1.5 Scope of the Project

Due to not having positions of ball at instantaneous periods, this method will slightly compromise on the accuracy of the results but it is negligible compared to the decrease in cost. As this technology is much cheaper compared to existing technology, this can be used at lower levels of cricket and other sports such as tennis and football. With the advent of HotSpot technology, this can be combined with HotSpot cameras to further reduce the cost.

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Ball Detection

2.1.1 Proposed Algorithm

The cricket ball is detected using Support Vector Machine (SVM) and Histogram of Oriented Gradient (HOG). Positive and negative data samples are collected and used for building the SVM models for HOG objects (ball, batsman). The video is sliced into multiple image frames separated by a fixed time duration. Frame difference technique is used to find areas which may have a high possibility of having a ball. Each window frame, after conversion to grayscale, is used to extract HOG features and fed to the SVM model to detect the ball in the areas found to be different in subsequent frames by the frame subtraction method.

Training Data Preparation

Data samples are collected by manually scraping through the captured videos and taking screenshots of relevant objects. There are two types of samples: negative and positive. Negative samples correspond to non-object images. Positive samples correspond to images with detected objects

Negative Samples

Negative samples are taken from arbitrary images. After taking a few negative samples, hard negative mining is applied, i.e., manually including all the false positive detections made by the SVM into the negative data sample set. A user interface was developed for the same which made the task easier and more efficient.

Positive Samples

Positive samples are collected by taking screenshots of a standardized size of the cricket ball from the captured video data. The positive sample set consists of a large number of images of a cricket ball differing in various parameters like orientation, color intensity, brightness, background, etc. The object instances are taken as screenshot images from the videos. Then they are resized to target samples size and stored in the positive sample set. No distortion is applied.

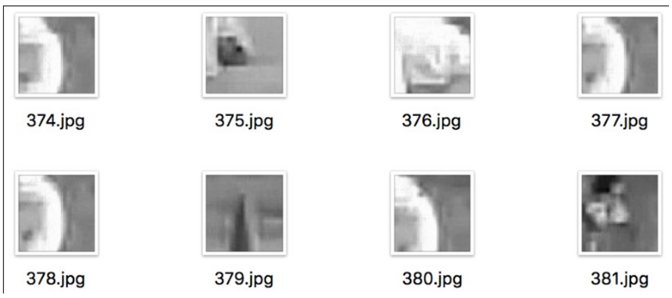

Figure 2.1: Negative Dataset

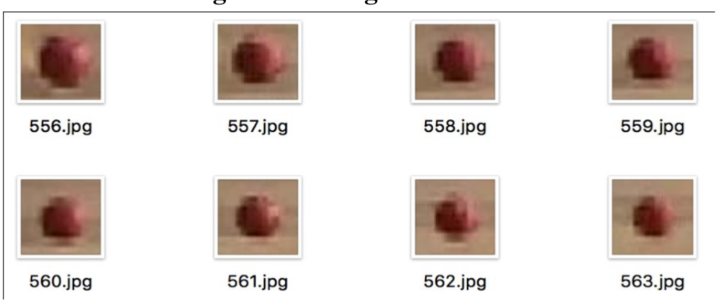

Figure 2.2: Positive Dataset

Histogram of Oriented Gradients (HOG)

OpenCV provides the functionality of extracting the HOG features of images using the various parameters mentioned below.

- - Minimum Window Size signifies the size of the window of which the HOG features are to be extracted.
- - Step Size indicates the amount by which the window moves in one iteration
- - Pixels Per Cell is the number of pixels that are there in a cell.
- - Cells per block is the number of cells in one block of the window.

Building the SVM

HOG features are extracted from the collected positive and negative data samples. The SVM model is built using the OpenCV SVM library using the extracted HOG features. As a result, we get an SVM model which is used to classify objects into those that are a ball and those that are not.

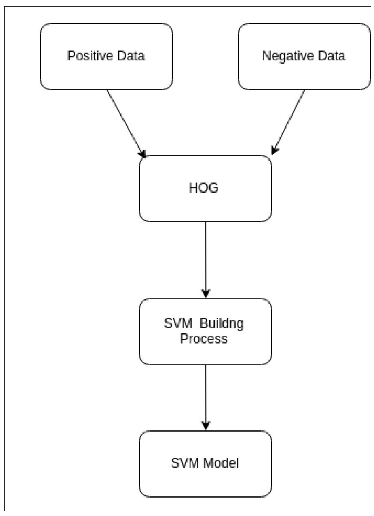

Figure 2.3: Building the SVM Model

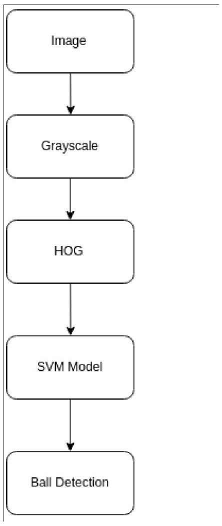

Figure 2.4: Ball detection process

2.2 Ball Tracking

Sliding Window Approach

The window where the ball is detected in a particular frame is used in the subsequent frame as a reference, and the SVM model is used to detect the presence of the ball in an inflated window in the neighborhood of the previous frame's window. This approach ensures higher accuracy in ball detection while improving performance.

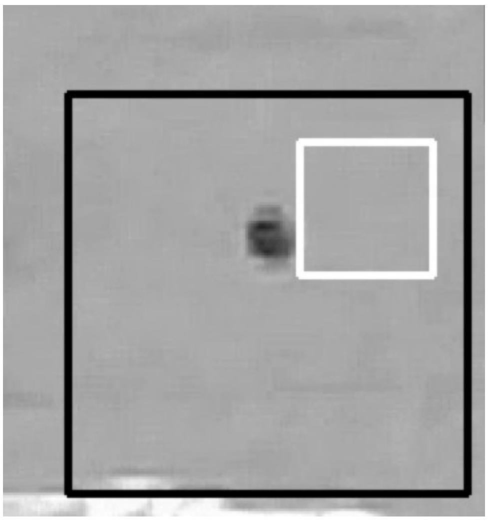

Figure 2.5: Sliding Window in progress

The obtained position of the ball is stored as a set of 'x' and 'y' coordinates in a separate file.

Bounce Point Detection

The point where the ball bounces is where the ‘y’ coordinate is the least.

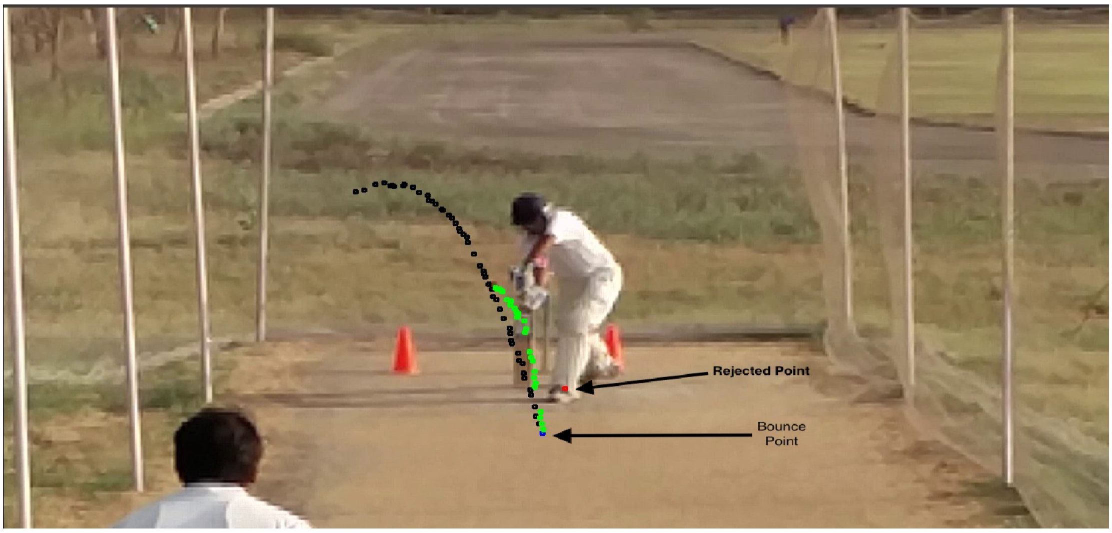

Figure 2.6: Ball Point Detection

Rejection of False Detections

Consecutive detections separated by a value less than a maximum distance are included in our correct detections. If the distance is greater than a particular value, the difference of angle between following points on the trajectory path followed by the ball is used, and false detections are rejected using a maximum permissible value of this angle. Further, some points are rejected while mapping to 3D coordinates based on the threshold values and the (x, y, radius) of the detected ball.

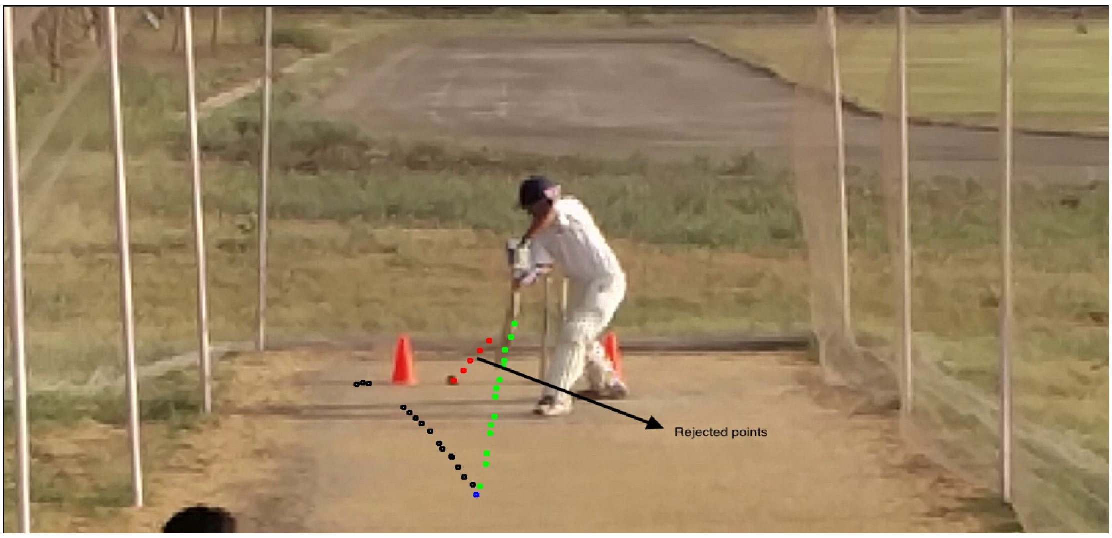

Figure 2.7: Red dots indicating rejected points

The above-mentioned maximum distance and angle difference methods are used to detect drastic changes in path due to the impact of the bat or batsman's body on the ball. All detections made after that would be false and hence are rejected.

Batsman Detection and Tracking

batsman is detected using an approach similar to that used for ball detection. OpenCV's built-in People Detector SVM models are used. Further, NMS is applied to the detected objects to obtain the final result. The movement of the batsman is tracked using a similar approach, i.e., limiting the search window to the neighborhood of the first detection. The detected batsman window is displayed on the frame.

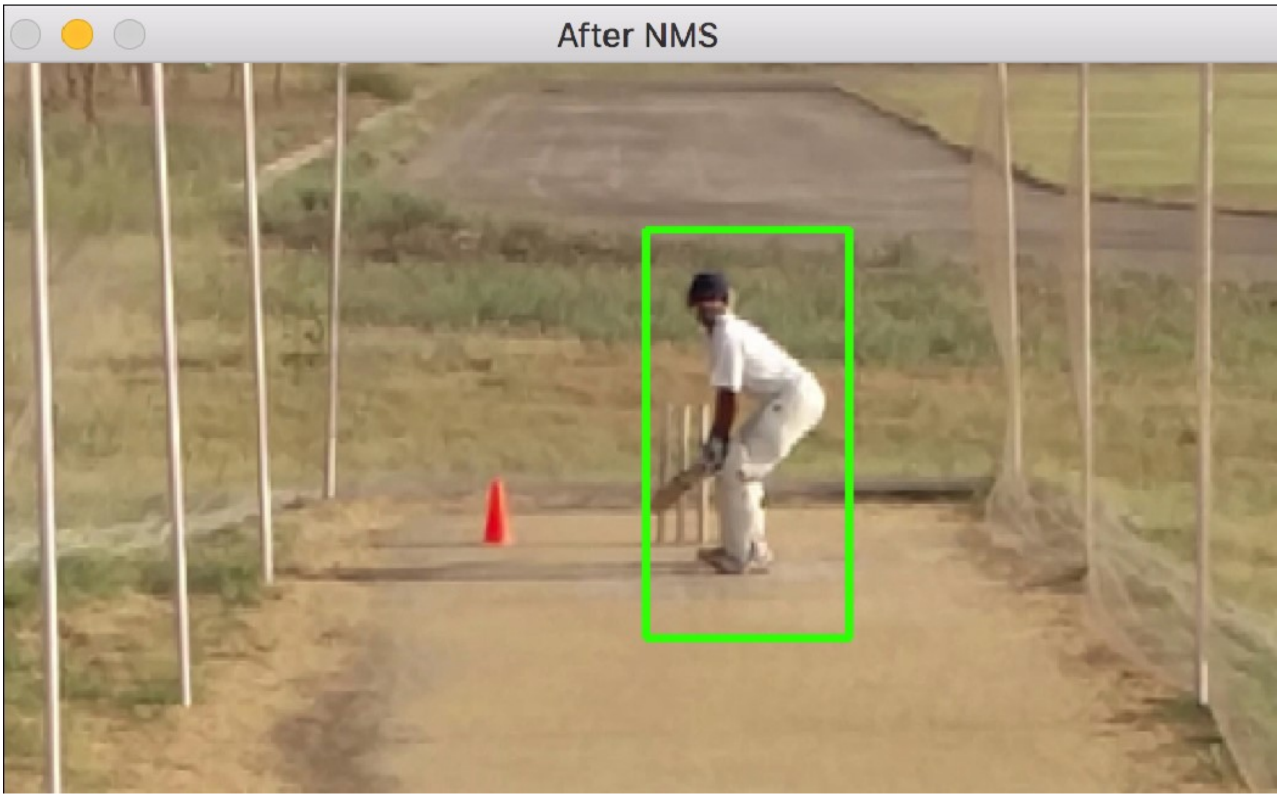

Figure 2.8: Tracked Batsmen after NMS

Umpiring Decisions

Information related to the ball and batsman is calculated using their obtained coordinates.

- The speed of the ball is found by taking an average of the speed of the ball before the bouncing point (if available) or after.

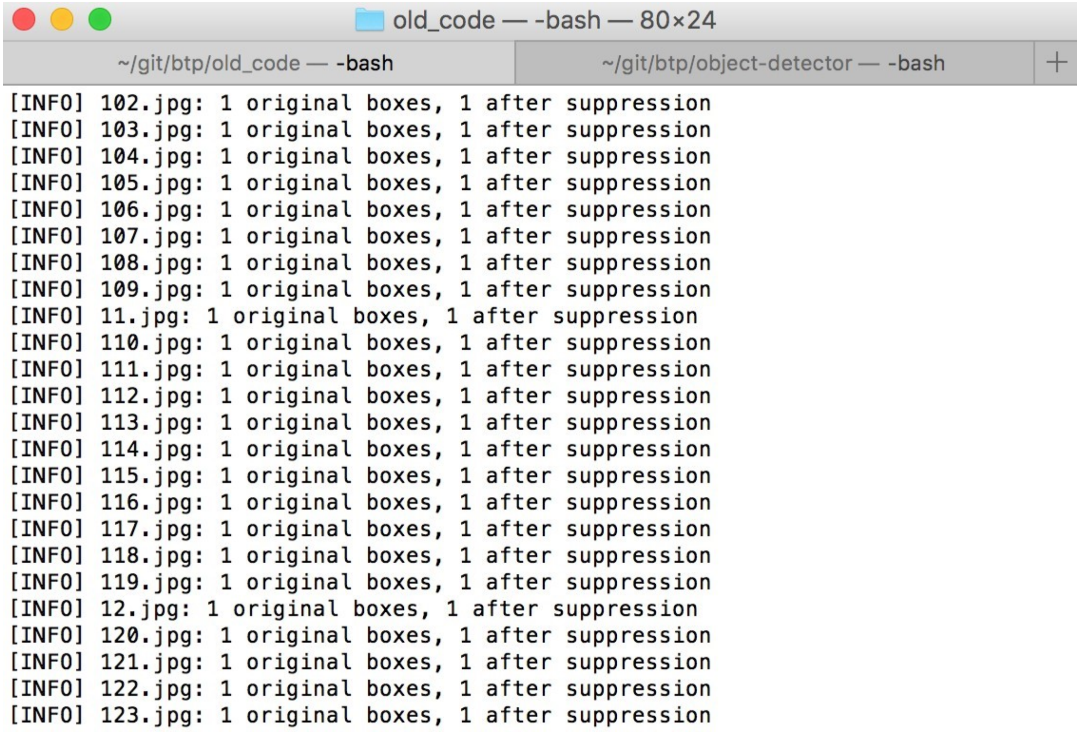

Figure 2.9: Batsman tracking logs across frames

- The height of the batsman is found by finding the midpoint of the detected batsman window, mapping the midpoint to real-world coordinates, and taking a double of the realworld height.

The obtained coordinates and other information is then used for umpiring decisions.

2.2.1 Leg Before Wicket (LBW) Decision

1. The impact location of the ball with the batsman is checked to see if it is inside the impact zone.
2. The pitching location of the ball is found.
3. The location of the ball when it is nearest to the wickets is found from the predicted ball trajectory, and this location is then checked to see if the ball would hit the wickets.

These three conditions are checked to arrive at the final LBW decision.

2.2.2 Wide Decision

1. The location of the ball when it is nearest to the batsman's wicket crease is found.
2. If the location is towards the leg side of the batsman, it is marked as Wide.

3. If the location is towards the off side of the batsman, the location is compared with the location of the Wide crease line.
4. If the height of the ball near the crease is greater than the height of the batsman, it is marked as wide.

2.2.3 No Ball Decision

1. The location of the ball when it is nearest to the batsman's wicket crease is found.
2. If the ball hasn't bounced off the ground, and the height of the ball is greater than half of the batsman's height, it is marked as a no-ball.

2.2.4 Bouncer Decision

1. The location of the ball when it is nearest to the batsman's wicket crease is found.
2. If the ball has a valid bounce point, and the height of the ball is greater than shoulder height of the batsman, it is marked as a bouncer.

Chapter 3

VISUALIZATION

3.1 3D Visualization of Tracked Objects and Cricket Pitch

Using the object (x, y, z) coordinates obtained by mapping algorithm, the objects are visualized in a 3D space using VPython visualization library.

- The pitch and the wickets are drawn by using boxes, taking ideal pitch measurements.
- The tracked ball is displayed on the screen using spheres with a cylindrical trajectory.
- The trajectory of the ball is predicted further using the polynomial equations obtained via regression and the predicted ball positions are displayed on the screen using spheres of a different color than the detected balls.
- The various decision parameters are shown on the screen.

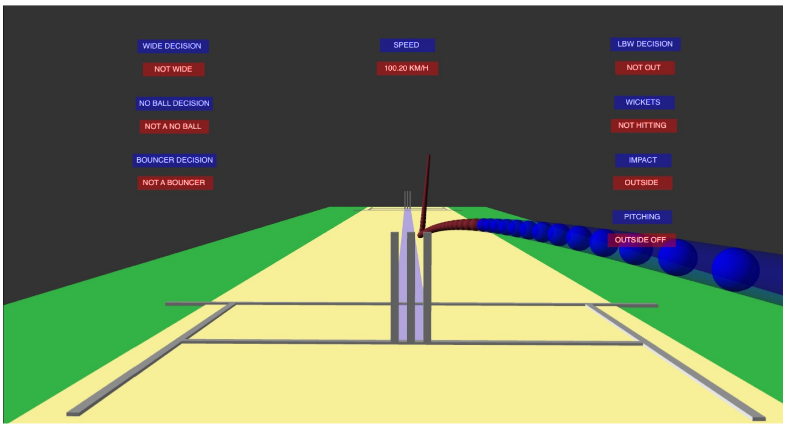

Figure 3.1: The visualization result

3.2 Graphical User Interface for the Application

A GUI for the system was developed using the Kivy python framework.

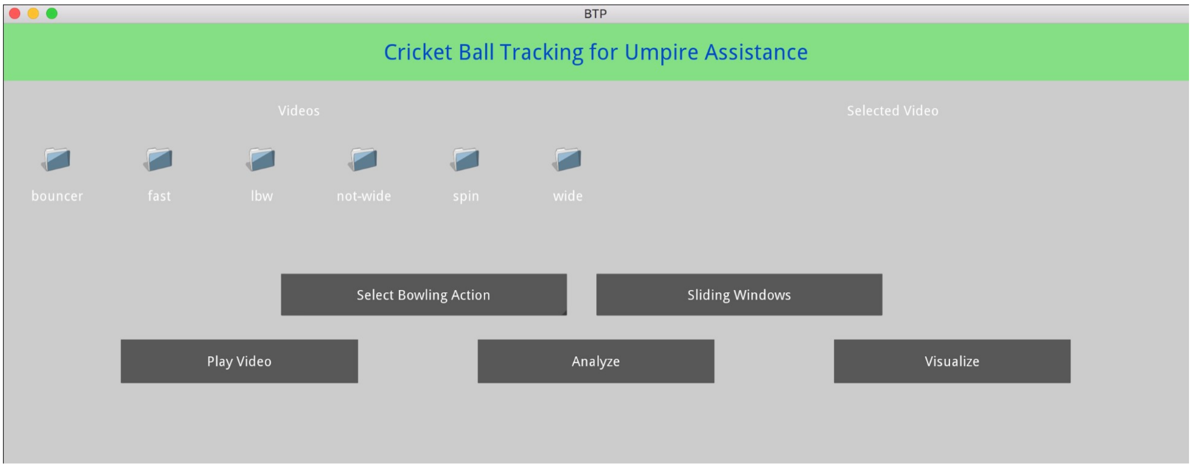

Figure 3.2: The GUI on first load

3.3 Implementation

- The Videos section allows selection of a video to analyze and visualize.
- The selected video's full name is displayed under the heading 'Selected Video'
- The Select Bowling Action dropdown button is used to choose the action between fast, slow and no-action for debugging purposes.
- The Sliding Windows toggle button when toggled shows the sliding windows in progress during ball detection.
- The Play Video button plays the selected video using a video player.
- The Analyze button activates the script that analyzes the video and finds the image coordinates of the tracked ball and player across frames.
- The Visualize button activates the script that visualizes the results of the analysis in 3D.

Figure 3.3: Video Analysis in progress

Chapter 4

EXPERIMENT AND CONCLUSION

4.1 EXPERIMENT

The videos used for developing and testing the system were taken from practice sessions of the JCC cricket team. Although a wide variety of videos were captured, 100+ in total, the size of the data wasn't enough for us to statistically determine the accuracy of the system. The system was later even deployed and used by the college team's coach during their practice sessions to improve player performance.

4.2 CONCLUSION

The primary objective of this paper was to develop a product for assisting the umpire in the sport of cricket while making decisions, using a single camera. It involved the development of algorithms using computer vision and machine learning techniques for ball detection and tracking, along with various cricket decision-making rules.
The paper discusses the use of computer vision to detect, identify and track the cricket ball (and other relevant objects in the context of cricket), and machine learning techniques to optimize and further predict various results and decisions. The use of just one camera, which may be of a quality equivalent to modern day smartphone cameras, along with the various algorithms and techniques of Computer Vision and Machine Learning helped us achieve a system that reliably assists the umpire and operates at a cheap cost.
The feature Histogram of Oriented Gradients (HOG) is implemented along with a Support Vector Machine (SVM) model to classify and detect the ball and batsman in a window frame. OpenCV and its various image processing features, with algorithmic techniques

like frame subtraction, sliding windows, CLAHE, minimum enclosing circle and machine learning techniques of weighted regression are implemented to achieve accurate tracking of the ball in motion.

Various umpiring decisions are made by checking specific conditions on the data obtained as per the rules of the sport of cricket and the decisions involved. VPython module is used to represent the decisions in the form of interactive 3D visualizations showing the ball trajectory path and information about the umpiring decisions. Kivy is used to provide an interactive graphical user interface to access and use the tool.

Chapter 5

FUTURE SCOPE

Due to not having positions of ball at instantaneous periods, this method will slightly compromise on the accuracy of the results but it is negligible compared to the decrease in cost. As this technology is much cheaper compared to existing technology, this can be used at lower levels of cricket and other sports such as tennis and football. With the advent of HotSpot technology, this can be combined with HotSpot cameras to further reduce the cost.

In this paper we attempted to come with a method to implement the projection of 3D objects in a 2D screen while paying enough stress at all the relative attributes of the object pertaining to its depth and its dimensions. The method places the image in a plane which is parallel to xz plane and the camera has the ability to move. The function thus developed for the projection considers the panning and uses concepts of trigonometry to calculate the revised coordinates. The possibilities of this implementation are numerous. One such attempt was its importance in LBW used in cricket to make correct decisions in case of close calls which might change the course of the match when there is a lot at the stake.

References

1. Bal, B. Dureja, G. (2012). “Hawk Eye: A Logical Innovative Technology Use in Sports for Effective Decision Making.” Sport Science Review, 21(1-2), pp. 107-119

2. Khan, Wazir Zada, Mohammed Y. Aalsalem, and Quratul Ain Arshad. ”The Aware Cricket Ground.” arXiv preprint arXiv:1109.6199 (2011)

3. Yan, F, Christmas, W and Kittler, J (2005). “A Tennis Ball Tracking Algorithm for Automatic Annotation of Tennis Match.” British Machine Vision Conference, 2005-09-05 - 2005-09-08, Oxford, UK.

4. Yu-Lin Chang, Wei-yinchen, Jing-Ying Chang, Yi-Min Tsai, Chia-Lin Lee and Liang-Gee Chen, ”Priority depth fusion for the 2D to 3D conversion system,” Proc

5. Fisher, J. “A line calling system to improve line umpire performance.” edited by Stuart Miller. International Tennis Federation, 2003.