# A Trajectory-Based Ball Detection and Tracking System with Applications to Shooting Angle and Velocity Estimation in Basketball Videos

Bodhisattwa Chakraborty
Dept. of Electronics and Communication Engg.
National Institute of Technology Rourkela
Odisha (769008), India
Email: bodhisattwa.chakraborty@gmail.com

Sukadev Meher
Dept. of Electronics and Communication Engg.
National Institute of Technology Rourkela
Odisha (769008), India
Email: sukadevmeher@gmail.com

***Abstract*—This paper presents a system to automatically analyse a basketball long-shot using trajectory-based ball tracking method from a basketball video sequence. The accuracy of a long-shot in a basketball game is mostly dependent on the ball throwing angle and the velocity at which the ball is to be thrown. The proposed system detects and tracks the ball in a basketball long-shot sequence by exploiting the trajectory information of the ball. The ball motion characteristics are used to determine the ball trajectory. The candidate trajectories are generated from a set of ball candidates in each frame. The trajectory-based ball tracking minimised the rate of error of ball tracking in the video which occurs due to occlusion and merging of the ball image with other objects in the frame, distortion of the ball image due to ball and camera motion and the presence of many moving objects in the foreground and background in the video. The ball locations verified by the tracking results are then used to estimate the ball throwing angle and the throwing velocity. The experiments show encouraging results for videos with dynamic background and different illumination conditions.**

***Index Terms*—Sports video analysis, detection-and-tracking, trajectory-based tracking, angle and velocity estimation.**

## I. Introduction

Basketball is one of the most popular games across the world having a great number of audiences. While common audiences intend to watch specific events rather than the entire game, the technical people associated with the game are interested in tactical analysis of the video which will help them to understand the tactics of the team and the performance of the players. Thus, automatic analysis of the basketball game videos becomes a burgeoning area of research in recent years.

The significant events in a basketball games are mainly caused by the interaction between the ball and/or the players. Thus, the detection and tracking of ball/players provide useful information for analysing the tactics and inferring the game strategy. Perše *et al.* [1] presented a trajectory-based method for automatic recognition of complex multi-player behaviour in a basketball game. The game is segmented in different phases by applying a probabilistic play model to the player trajectory and the game is analysed by detecting the key elements of the basketball play. Fu *et al.* [2] proposed a screen-strategy analysis method in basketball videos using the player tracking. The player trajectories are computed by a Kalman filter based tracking method. The shooting location estimation system using the ball trajectory in a basketball game has been proposed by Chen *et al* [3]. A physics-based algorithm is used to track the ball and the 3-D trajectories are reconstructed from the extracted 2-D trajectories. Liu *et al.* [4] presented a shot identification method in basketball video based on ball tracking. The ball tracking was achieved by mean-shift algorithm and Kalman filter and the hoop was identified using SURF. In our previous work, we have successfully applied a trajectory-based approach to detect and track the ball in basketball videos [5].

In this paper a ball detection-and-tracking algorithm based on trajectory-based ball tracking for basketball videos is proposed. The estimated ball locations from the ball trajectory are used to calculate the ball throwing angle and the velocity of throw. The main advantage of the proposed algorithm is it can be used to analyse the basketball long shot sequence irrespective of the illumination conditions and the surrounding environments. The hardware requirement for the algorithm is minimal as it requires only a single camera to capture the video. The computational complexity of the algorithm is also very less, thus providing a real-time analysis of the basketball long-shot sequences.

The rest of the paper is organized as follows. The proposed model is discussed in section II. An overview of the process of moving object segmentation is presented in section III. Section IV presents the method of ball candidate identification using shape and circularity constraints. The candidate trajectory generation process and the ball trajectory identification process are discussed in section V. In section VI the process of shooting angle and velocity estimation of the ball is presented. Experimental results are analysed in section VII and finally the paper is concluded in section VIII.

## II. Proposed System

The flowchart of the proposed framework for ball detection-and-tracking in a basketball video and its analysis is shown in

Fig. 1. The framework starts with a moving object segmentation method in the video sequence to separate the moving objects from the dynamic background using a robust model comprising of frame differencing and background subtraction. In the second step, a set of ball candidates is generated using feature-based method for each frame. The set of candidate trajectories are generated from the ball candidates and the true ball trajectory is identified by exploiting the physical characteristics of ball motion along the x- and y- direction separately. The trajectory-based ball tracking methods also helps to predict the missing ball locations in the frames by employing an interpolation technique. The ball locations verified by the ball trajectory are then used to estimate the ball throwing angle and the ball throwing velocity for a basketball long-shot. The proposed framework is tested with videos of different resolutions and yields excellent results.

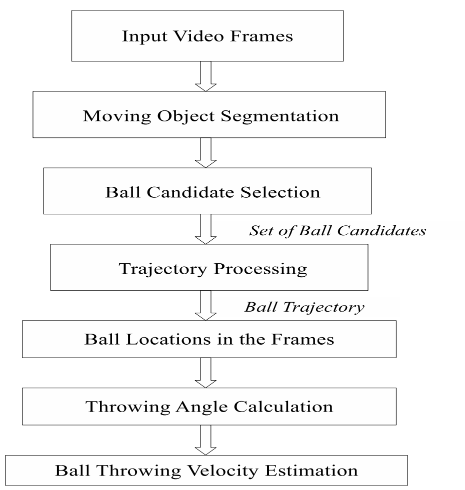



Fig. 1: Flowchart of the proposed model for Basketball video analysis.

## III. MOVING OBJECT SEGMENTATION

The basketball video for indoor basketball courts normally contains many moving objects in the background such as flags, banners, spectators etc. whereas for outdoor basketball courts, the waving of twigs and leaves of trees generates dynamic background. The illumination condition keeps varying in case of indoor and outdoor basketball court and for videos taken during different times (for example videos taken during day time and night time) in a day. To overcome these situations, a robust moving object segmentation algorithm is required which can withstand the effects of illumination changes and the presence of multiple moving objects in the background. In this paper, a background subtraction method along with a frame differencing method using three consecutive frames are employed for segmentation of the moving objects in the foreground [6]. The first frame of the input video is considered as the background frame $B(x, y)$ and the subsequent frames are subtracted from the background frame. For frame $I_i(x, y)$, the background subtraction yields a difference image $d_B(x, y)$.

$$d_B(x, y) = |I_i(x, y) - B(x, y)| \tag{1}$$

The frame differencing is performed on three consecutive frames between frames $I_i(x, y)$ and $I_{i-1}(x, y)$ and between frames $I_i(x, y)$ and $I_{i+1}(x, y)$. This results in two difference images $d_{i-1}(x, y)$ and $d_{i+1}(x, y)$ which are thresholded to generate two binary masks as shown by Eq. 2 and Eq. 3 respectively.

$$\begin{aligned} d_{i-1}(x, y) &= I_i(x, y) - I_{i-1}(x, y) \\ d_{i+1}(x, y) &= I_i(x, y) - I_{i+1}(x, y) \end{aligned} \tag{2}$$

$$d_k(x, y) = \begin{cases} 1, & \text{if } d_k(x, y) > T \\ 0, & \text{Otherwise.} \end{cases} \tag{3}$$

where, $k = i - 1$ and $i + 1$ and $T$ is a threshold ranging form 0.43 to 0.6 for different video sequences based on the average intensity of the frames. The same threshold is applied to $d_B(x, y)$ to generate the binary mask $d_{B'}(x, y)$ of the difference image. Logical *AND* operation is performed between $d_{B'}$ and $d_{k-1}$ and between $d_{B'}$ and $d_{k+1}$ to generate two motion masks ${m^p}_{i-1}$ and ${m^p}_{i+1}$ respectively.

$$\begin{aligned} {m^p}_{i-1} &= d_{B'} \cap d_{k-1} \\ {m^p}_{i+1} &= d_{B'} \cap d_{k+1} \end{aligned} \tag{4}$$

Lastly, the motion mask ${m^p}_i$ for frame $I_i(x, y)$ is generated by applying an *OR* operation between the two motion masks derived in Eq. 4.

$${m^p}_i = {m^p}_{i-1} \cup {m^p}_{i+1} \tag{5}$$

The background is updated to tolerate the effect of illumination variation and background clutter if the percentage of the pixels in motion is less than a threshold. Otherwise, the background remained unchanged.

$${n^m}_i = \frac{n^m}{M \times N} \times 100\% \tag{6}$$

$$B(x, y) = \begin{cases} I_i(x, y), & \text{if } {n^m}_i < T_m \\ B(x, y), & \text{Otherwise.} \end{cases} \tag{7}$$

where, ${n^m}_i$ is the percentage of pixels in motion for frame $I_i(x, y)$, $n^m$ is the number of pixels in motion in the motion mask ${m^p}_i$, $M \times N$ denotes the total number of pixels in the frame $I_i(x, y)$ and $T_m$ is the motion threshold. The value of motion threshold $T_m$ is set to 4%. After the moving object segmentation, the morphological opening and closing operations [7] are performed on the derived motion mask to remove the noises that are created during the segmentation and the camera motion. While morphological opening operation soothes the contour of the object and breaks thin projections due to motion discontinuity, the closing operation joins the narrow brakes and fills the holes in the motion mask.

## IV. BALL CANDIDATE SELECTION

In a basketball video, the presence of players generates a number of moving objects in the foreground of the video frames. The dynamic background comprising of flags, banners, spectators, twigs and leaves of branches often leads to wrong segmentation of the scene despite of a robust moving object segmentation algorithm used. The high speed motion of the ball and the camera motion deform the ball image. The merging of the ball image with other objects in the frame and the occlusion of the ball with players also leads to the deformation of the ball image by a great extent. To filter out the original ball image from other moving objects present in the frame, some feature-based filters has to be used. In this work, the shape and circularity features of the ball are used to distinguish the ball image. The objects that do not satisfy the shape and circularity constraints are pruned and the remaining objects are considered as "ball candidates" in the frame.

### A. Shape Filter

The shape of the ball changes drastically in a basketball game video as an effect of motion blurring. The motion blurring is mainly caused by the motion of the ball as well as the camera motion. The occlusion and merging of the ball image also leads to the deformation of the ball image. It has been seen that the ball image has more resemblance with an ellipse than a circle in video frames. Thus the *eccentricities* of the object blobs are used to determine the shape of the object. The *eccentricity* of a blob is calculated as the ratio of the height and width of the smallest rectangle containing every point in the shape. The eccentricity of a circle is 0 and that of an ellipse is greater than 0 but less than 1. The threshold of the shape filter is selected as 0.7.

### B. Circularity Filter

In a basketball video sequence, many objects resemble a ball and often the non-ball objects looks like a ball more than the ball itself. These objects can pass through the shape filter and lead to wrong detection. Thus a filter based on the circularity of an object in the frame is employed. Circularity can be defined as the ratio of the area of a region to the area of a circle having the same perimeter. In this paper, we have used *normalised circularity* [8] of the object blobs to find out the ball candidates in the frame. The normalised circularity can be defined as,

$$\gamma_N = 1 - \frac{4\pi A}{P^2} \tag{8}$$

where, $A$ is the area of the blob and $P$ is the perimeter of the blob. The value of $\gamma_N$ for a circle is 0 and 1 for all other complex shapes. Thus the blobs with $\gamma_N$ value in the range $0 < \gamma_N < 0.5$ are retained as ball candidates.

## V. TRAJECTORY PROCESSING

To tolerate the shape and compactness variations in near and far view frames, a wide range of thresholds are used for the filters. As a result some non-ball objects pass through the filters and miss-classified as ball candidates. Thus, the ball motion characteristics along x-direction and y-direction over a number of frames are used to determine the actual ball locations. The ball locations along x- and y-direction are plotted against the number of frames to generate the x-candidate plot (XCP) and y-candidate plot (YCP) respectively as shown in Fig. (2a) and (2b). It has been observed that the ball moves in a straight line along the x-direction and follows a near parabolic path along y-direction. This information is used to generate the candidate trajectories in XCP and YCP.

### A. Candidate Trajectory Generation

The candidate trajectory generation algorithm starts with a pair of ball candidates in consecutive frames which are close to each other. A Kalman filter-based [9] prediction method is used to predict the ball locations along the trajectory and it is independent of velocity and acceleration of the ball. The Kalman filter-based system can be describes as,

$$\begin{aligned} x_i &= A.x_{i-1} + w_{i-1} \\ z_i &= H.x_i + v_i \end{aligned} \tag{9}$$

where $x_i$ is the state vector representing the estimated ball location in $i^{th}$ frame and $z_i$ is the measurement vector which is the position of detected ball candidate. $A$ is the system evolution matrix and $w$ is the process noise vector. $H$ is the measurement matrix and $v$ represents the measurement noise vector. The initial system evolution matrix and the measurement matrix are taken as,

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

The predicted ball location in each frame is verified by the ball candidate in that frame. If the predicted location is verified, the prediction function is updated and the trajectory growing process is continued. If the predicted location is not verified, the Kalman filter continues to predict the ball location until the number of frames for which the prediction is not ($N_{miss}$) verified exceeds a threshold ($T_{miss}$). The value of $T_{miss}$ is set to 05 frames. The candidate trajectory generation algorithm is described in Algorithm 1. The candidate trajectories generated by Algorithm 1 for XCP and YCP are shown in Fig. (2c) and (2d) respectively.

### B. Ball Trajectory Identification

The actual ball trajectory has been identified from the set of candidate trajectories using two criterions: i) the trajectory length ($L_{traj}$) and ii) the prediction error ($e_p$). The ball in

**Algorithm 1** Candidate Trajectory Generation

**Input:** Set of ball candidates
**Output:** Set of candidate trajectories
**for** each frame in video **do**
  **for** each ball candidate $b_t$ in frame $t$ **do**
    **for** each ball candidate $b_{t+1}$ in frame $(t+1)$ **do**
      **if** distance($b_t, b_{t+1}$)$< d_{min}$ **then**
        Initialise the Kalman filter;
        Predict the location for frame $(t+2)$;
        **if** the prediction is verified **then**
          Add $b_t$, $b_{t+1}$ and $b_{t+2}$ to trajectory $T_{cand}$;
          Update prediction function;
        **else**
          **if** $N_{miss} > T_{miss}$ **then**
            Record new trajectory;
          **end if**
          Estimate new ball location;
        **end if**
      **end if**
    **end for**
  **end for**
**end for**

a basketball long shot sequence moves continuously for a number of consecutive frames. Thus the ball trajectory should be the longest among the set of candidate trajectories. It has been observed that for a basketball long shot sequence, the ball remains airborne for almost 15 to 18 consecutive frames. Thus the threshold of the trajectory length ($T_L$) is empirically set to 10 frames for this work. The prediction error is defined as the average distance (in pixel) between each predicted location to the ball candidate location in a frame. The candidate trajectories having prediction error greater than a threshold ($T_p$) are eliminated. The value of $T_p$ is selected to be 05 (in pixel) for this work. The process for ball trajectory identification is shown in Algorithm 2.

**Algorithm 2** Ball Trajectory Identification

**Input:** Set of candidate trajectories $S_T(F)$
**Output:** Ball trajectory $B_T$
**for** a candidate trajectory $T_{cand}$, $T_{cand} \in S_T(F)$ **do**
  **if** $T_{cand}$ forms a line in XCP and a parabolic curve in YCP **then**
    **if** ($L_{traj} \geq T_L$ && $e_p < T_p$) **then**
      $T_{cand} \rightarrow B_T$
    **else**
      Remove $T_{cand}$ from $S_T(F)$
    **end if**
  **end if**
**end for**

The ball trajectory identified by Algorithm 2 in XCP and YCP are shown in Fig. (2e) and (2f) respectively.

## VI. APPLICATIONS

The throwing angle and velocity of the ball defines the accuracy of a long shot in a basketball video. Thus it becomes absolutely necessary to calculate these two parameters to analyse a long shot sequence. Once the ball locations are identified in the frame using the trajectory information, the throwing angle and the velocity of the ball can be easily calculated.

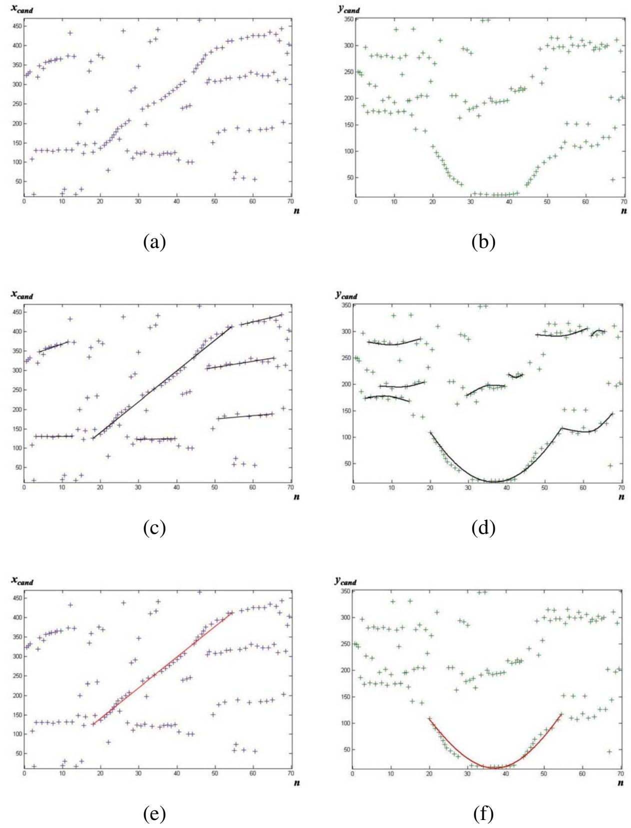



Fig. 2: Illustration of candidate trajectory generation and ball trajectory identification using x- and y-distribution analysis.

### A. Ball Throwing Angle Calculation

The ball throwing angle ($\theta_{throw}$) can be calculated as,

$$\theta_{throw} = tan^{-1}\frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)} \tag{11}$$

where, $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$ are the coordinates of ball location for $i^{th}$ and $(i+1)^{th}$ frame respectively.

### B. Ball Throwing Velocity Estimation

The ball throwing velocity can be calculated as,

$$v_{throw} = 4.9 \cdot \frac{t_{throw}}{sin(\theta_{throw})} \quad m/s \tag{12}$$

where, $t_{throw}$ is the ball flight duration and can be calculated as,

$$t_{throw} = \frac{T_L}{f_r} \quad seconds \tag{13}$$

where, $T_L$ is the length of the ball trajectory and $f_r$ is the frame rate of the video.

TABLE I: Performance of the proposed system for ball detection and tracking

| Video Clips | **Ground Truth** | | **Ball detection result** | | | **Ball tracking result** | | |
|---|---|---|---|---|---|---|---|---|
| | *Total frames* | *Ball frames* | *Correct* | *False* | *Accuracy (%)* | *Correct* | *False* | *Accuracy (%)* |
| BBD-0005 | 70 | 25 | 59 | 11 | 84.29 | 62 | 08 | 88.57 |
| TPB-0002 | 90 | 29 | 84 | 06 | 93.33 | 87 | 03 | 96.67 |
| BBSD-0001 | 50 | 32 | 41 | 09 | 82.00 | 44 | 06 | 88.00 |
| BBN-0022 | 60 | 31 | 55 | 05 | 91.67 | 57 | 03 | 95.00 |
| B3P-0003 | 65 | 36 | 55 | 10 | 84.62 | 61 | 04 | 93.85 |
| BBN-0020 | 90 | 27 | 76 | 14 | 84.44 | 85 | 05 | 94.44 |
| **Total** | 425 | 180 | 370 | 55 | 87.06 | 396 | 29 | 93.18 |

## VII. EXPERIMENTAL RESULTS

The proposed algorithm for ball detection-and-tracking with applications to shooting angle and velocity estimation is tested with a set of six videos. The test videos are having different resolution (360p, 480p, 720p) and the illumination conditions are also varying. Some videos are of indoor basketball court, while some are of outdoor court environment, thus providing a wide range of variety in the background scenes. The ground truth ball locations are detected using ViPER (Video Performance Evaluation Resource) video annotation tool [10]. The results of ball detection and tracking are summarized in Table I. "*Correct*" refers to those frames where the algorithm detects and tracks a ball in a ball frame and does not detect and track a ball in a non-ball frame where "*False*" refers to those frames where a ball is wrongly detected in a non-ball frame and/or the algorithm fails to detect and track the ball in a ball frame. The accuracy of ball detection and tracking is calculated as the ratio between the correct detection to the number of total frames in the video. It can be seen that, the proposed algorithm successfully detects a ball in the basketball long shot test video sequences with an average accuracy of 87.06%. The use of trajectory-based method improves the result by a great extent and the final result of ball detection-and-tracking yields and average accuracy of 93.18%.

TABLE II: Results of ball throwing angle and velocity estimation

| **Video Sequence** | $f_r$(in fps) | $t_{throw}$(in sec) | $\theta_{throw}$ (in degree) | $v_{throw}$ (in meter/sec) |
|---|---|---|---|---|
| BBD-0005 | 29 | 0.8621 | 84.47 | 4.24 |
| TPB-0002 | 25 | 1.16 | 53.62 | 7.06 |
| BBSD-0001 | 25 | 1.1034 | 55.18 | 6.59 |
| BBN-0022 | 29 | 1.07 | 68.73 | 5.60 |
| B3P-0003 | 30 | 1.20 | 58.64 | 6.86 |
| BBN-0020 | 29 | 0.931 | 73.19 | 4.77 |

In Table II the results of shooting angle estimation and ball throwing velocity estimation using the proposed method has been shown. In this work, air resistance to the movement of the ball is considered negligible. Also, the ball release height is not considered here.

Fig. 3 shows the experimental results of ball detection-and-tracking in a set of basketball videos. The first row shows the results of ball detection solution while the second row shows the results of trajectory-based solution where the missing ball locations are predicted and verified. The detected and tracked ball locations are shown by green dots. The interpolated ball locations during trajectory processing are represented by yellow dots.

To compare the performance of the proposed algorithm, a mean-shift based tracking method [11] is implemented. The mean shift algorithm is a well-known statistical method for finding local maxima in probability distributions which is extensively used in the field of object tracking. For performance evaluation, the track detection rate (TDR) and the false alarm rate (FAR) are used. TDR and FAR can be derived as,

$$TDR = \frac{TP}{TP + FN}$$
$$FAR = \frac{FP}{TP + FP} \tag{14}$$

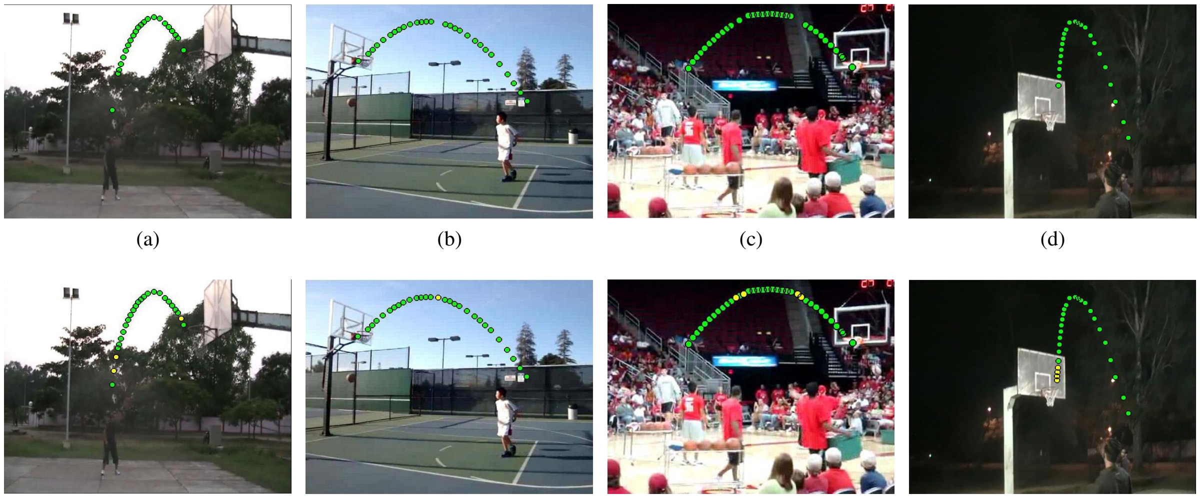
where, '*TP*' is the number of true positives for the tracked object, '*FP*' is the number of false positive and '*FN*' is the number of false negative. The comparison results are shown in Table III. It can be observed that the average TDR for the proposed method is as high as 95.29% where that of the mean-shift based method is 67.22%. The FAR of the proposed method is very less (18.59%) as compared to the mean-shift based method (29.65%). Fig. 4 illustrates the comparison on ball tracking in terms of TDR and FAR.

TABLE III: Comparison of proposed method and Mean-Shift based method

| Video Clips | **Proposed Method** | | **Mean-Shift based Method** | |
|---|---|---|---|---|
| | *Track Detection Rate (%)* | *False Alarm Rate (%)* | *Track Detection Rate (%)* | *False Alarm Rate (%)* |
| BBD-0005 | 88.00 | 26.67 | 44.00 | 47.62 |
| TPB-0002 | 96.55 | 15.15 | 75.86 | 29.03 |
| BBSD-0001 | 87.50 | 15.15 | 71.87 | 23.33 |
| BBN-0022 | 93.55 | 9.38 | 70.97 | 18.52 |
| B3P-0003 | 88.89 | 15.79 | 63.88 | 32.35 |
| BBN-0020 | 85.19 | 30.30 | 74.07 | 31.03 |
| **Total** | 95.29 | 18.59 | 67.22 | 29.65 |

## VIII. CONCLUSIONS

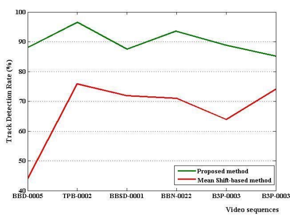
In this paper, a trajectory-based ball detection-and-tracking framework is presented which can be used to extract the ball locations in basketball long shot video sequences. The motion characteristic of the ball is used to identify the ball trajectory using 2-D distribution analysis of the ball candidates along x- and y-direction separately. A robust moving object



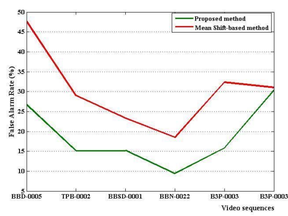
(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3: Illustration of ball position estimation using the trajectory information for different basketball long shot video sequences.




(a)




(b)

Fig. 4: Comparison of ball tracking between proposed method and mean-shift based method. (a) TDR plot, (b) FAR plot.

segmentation method based on background subtraction and frame differencing is used to determine the moving objects in the foreground which gives reliable results for dynamic background scenes with heavy background clutter and can withstand the effects of camera motion. The feature-based pruning of the ball candidates ensures a less number of candidates to be processed during the trajectory processing which reduces the computational complexity of the overall system. It also reduces the FAR which leads to an excellent TDR for the proposed method. The extracted ball locations are used to determine the throwing angle of the ball which in turn is used to determine the shooting velocity. The system is very much cost effective as it does not require sophisticated hardware like high speed cameras. We believe this is the first algorithm where a complete analysis of a basketball long shot sequence is presented in real-time basis.

The same trajectory-based approach can be used to detect and track the ball in other sports videos in which the ball moves in similar motion characteristics. We have implemented a similar trajectory-based method to track the ball and recognise the set types in volleyball videos [12]. In future, we are looking to develop a system which can be used for further analysis of the sports video taking into consideration about the other factors like air friction, ball spin etc. 3-D trajectory reconstruction will be used to get more information about the ball trajectory and for better representation. The same approach can be used in surveillance application where the trajectory of the moving object has to be tracked.

## References


[1] M. Perse, M. Kristan, S. Kovacic, G. Vuckovic, and J. Pers, "A trajectory-based analysis of coordinated team activity in a basketball game," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 612 – 621, 2009.

[2] T.-S. Fu, H.-T. Chen, C.-L. Chou, W.-J. Tsai, and S.-Y. Lee, "Screen-strategy analysis in broadcast basketball video using player tracking," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, nov. 2011, pp. 1 –4.

[3] H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, and S.-Y. Lee, "Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video," *Journal of Visual Communication and Image Representation*, vol. 20, no. 3, pp. 204–216, 2009.

[4] Y. Liu, X. Liu, and C. Huang, "A new method for shot identification in basketball video," *Journal of Software*, vol. 6, no. 8, pp. 1468–1475, 2011.

[5] B. Chakraborty and S. Meher, "A real-time trajectory-based ball detection-and-tracking framework for basketball video," *Journal of Optics*, vol. 42, no. 2, pp. 156–170, 2013.

[6] S. M. Desa and Q. A. Salih, "Image subtraction for real time moving object extraction," in *International Conference on Computer Graphics, Imaging and Visualization (CGIV 2004)*, 2004, pp. 41–45.

[7] F. Y. Shih, *Image Processing and Mathematical Morphology: Fundamentals and Applications*. CRC Press, 2009.

[8] S. K. Mitra and G. L. Sicuranza, *Nonlinear Image Processing*. Academic Press, 2001.

[9] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

[10] "Viper - video performance evaluation resource, http://viper-toolkit.sourceforge.net/owl/viper/datatypes/."

[11] J.-X. Pu and N.-S. Peng, "Adaptive kernel based tracking using mean-shift," vol. 4141, pp. 394–403, 2006.

[12] B. Chakraborty and S. Meher, "A trajectory-based ball detection and tracking system with applications to shot-type identification in volleyball videos," in *IEEE International Conference on Signal Processing and Communications (SPCOM)*, 2012, pp. 1 –5.