Automated Ball Tracking in Tennis Videos

Tayeba Qazi[1], Prerana Mukherjee[2], Siddharth Srivastava[3], Brejesh Lall[4], Nathi Ram Chauhan[5]
[1,5]Department of Mechanical and Automation Engineering
Indira Gandhi Delhi Technical University for Women Delhi, India
[2,3,4]Department of Electrical Engineering, Indian Institute of Technology, Delhi, India
[1,5]qazitayeba,nnramchauhan@gmail.com, [2,3,4]{eez138300, eez127506, brejesh}ee.iitd.ac.in

***Abstract*—Sports video analysis tools are gaining enormous popularity as they enable enhanced visualization and analysis of the game. An intriguing problem is localizing and tracking the ball in a game of tennis. However, achieving a practically suitable trade-off between high detection accuracy and speed of the ball is a challenging problem in automatic ball tracking algorithms. In this paper, we propose a machine learning based automated ball tracking algorithm in tennis match videos acquired by a camera mounted on a quadcopter. We begin with applying a video stabilization technique followed by random forest segmentation for detecting tennis ball candidates in the video frames. The yellow color plane intensity and Phase Quaternion Fourier Transform (PQFT) saliency are used as features for random forest segmentation. The effectiveness of the proposed algorithm is demonstrated on datasets from real-world tennis videos. The experimental results show that our algorithm is robust and achieves 94% accuracy in best case and successfully detects the ball even in cases of partial occlusion by net or racket.**

Keywords*- *Tennis video, Ball Tracking, Video Stabilization, Random Forest Segmentation

I. Introduction

Tennis is one of the most popular sports across the globe. A meticulous analysis of the game is needed to reduce human errors and extract several statistics from the visual feed of the game. The advent of computer vision and machine learning techniques have effectively replaced manual analysis of sports with fully automated systems. Automated ball tracking is one such class of systems which requires sophisticated algorithms for analysis. Nowadays, cameras mounted on quadcopters are used to capture such videos which makes the problem even more difficult. The challenges arise as small sized quadcopters are susceptible to atmospheric turbulence and rotor vibrations making the video jittery. Moreover, small size of the tennis ball (67mm diameter), high ball speed (fastest 225kmph), variation in illumination and contrast conditions, multiple objects in motion in the same frame and presence of multiple objects with similar attributes presents additional challenges.

Various techniques have been developed for automated tennis ball tracking which address one or more of the above discussed challenges. Broadly, a tennis ball tracking algorithm involves five steps. First is ball detection using frame difference [1], frame subtraction [2], template matching [3], morphological operations [4] or anti-modal method [5]. Second is classification which takes shape and color information [6]. Third is performing logical AND operation between frames and masks of frames [7], [8]. Fourth is ball extraction which involves blob analysis based on shape, size or color of the ball [3], [1], [5], [4], [2]. Recent works perform foreground blob classification using features for detecting elliptical objects [9], LMR algorithm [8], hue-saturation model [1]. Fifth is Ball Trajectory Generation involving position prediction based on previous displacement [3], particle filter [9], 2D motion model [10].

In this paper we develop a robust efficient ball tracking system. We merge the strengths of computer vision and machine learning techniques and present an algorithm which scores better than the existing algorithms in terms of accuracy and speed. The key idea is to define a video stabilization framework followed by segmentation of ball candidates in the video using random forest classification. We exploit the saliency map of the frame along with the color of the ball for effective ball candidate segmentation in video frames.

In the subsequent section II, we discuss the proposed methodology, the experimental results are provided in section III and the conclusion and future work is given in section IV.

II. Proposed Approach

In this Section, we give a detailed overview of the automatic tennis video annotation framework. The workflow of the proposed method is shown in Figure 1. In the following subsections we describe the components in detail.

A. Video Stabilization

The video of the tennis match acquired by the quadcopter has extreme jerks and noise, hence requiring stabilization before further processing. We have used the video stabilization framework proposed in [11] to stabilize the video of the tennis match and do further processing to track the ball. The steps involved in video stabilization are: (i) Feature detection and matching (ii) Homography Estimation (iii) Parameter Smoothing (iv) Frame Warping as shown in Figure 2 and are discussed next.

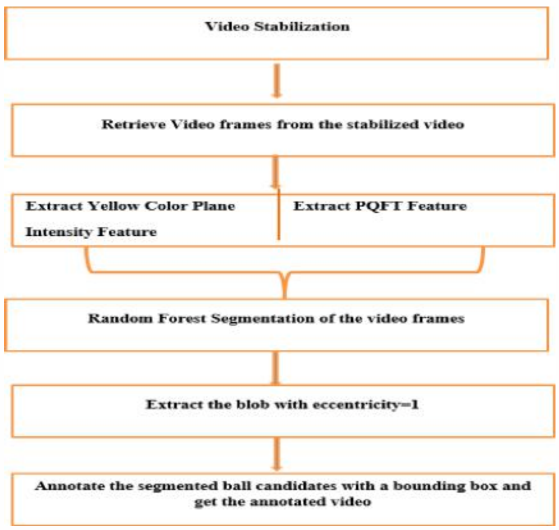

Fig. 1. Ball Tracking Framework

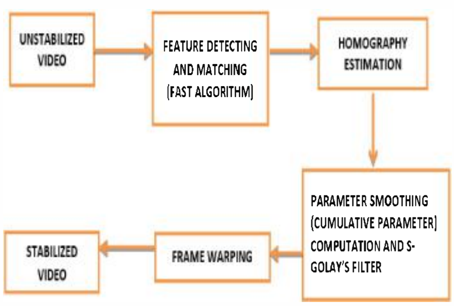

Fig. 2. Video Stabilization Framework

1) Feature Detection and Matching: We use the FAST corner detection algorithm [15] for finding keypoints in various frames which is time and memory efficient method to compute the features for video stabilization. Then we extract Fast Retina Keypoint (FREAK) descriptor centered around the FAST keypoints (Figure 3). The corner key points of each frame are matched with the corresponding corner key point in the subsequent frame using the concept of Hamming Distance. In the figure 3, the key points of current frame are marked in green and the key points of the subsequent frame in red. The corresponding shift of each key point due to unwanted motion in the video is calculated and depicted by yellow lines.

2) Homography Estimation: After the feature matching step, we perform homography estimation between the point correspondences in both the frames. In a homography transform eight parameters are encoded, which estimates the translation, rotation, scaling, skew, and perspective transformation occurring in a point as below.

$$x' = Hx \tag{1}$$

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2}$$

where x is a point in frame A, x' is a point in frame B, H is the estimated homography. Matrix H is given by,

$$H = \begin{bmatrix} s.acos\Theta & s.bsin\Theta & t_x \\ s.asin\Theta & s.bcos\Theta & t_y \\ P_x & P_y & 1 \end{bmatrix} \tag{3}$$

where s is the scaling factor, a and b are the skew factors, $sin\theta$ and $cos\theta$ are the rotation factor, t_x and t_y are the translation factor, P_x and P_y are the perspective transform.

3) Parameter Smoothing: We combine the obtained transform which describes all camera motion after the first frame, to obtain a cumulative transform with respect to the first frame. Subsequently, SavitzkyGolay filter [12] is applied for the purpose of smoothing the motion parameters so obtained.

4) Frame Warping: Finally the motion parameters so computed are warped to each frame in order to obtain a stabilized the video.

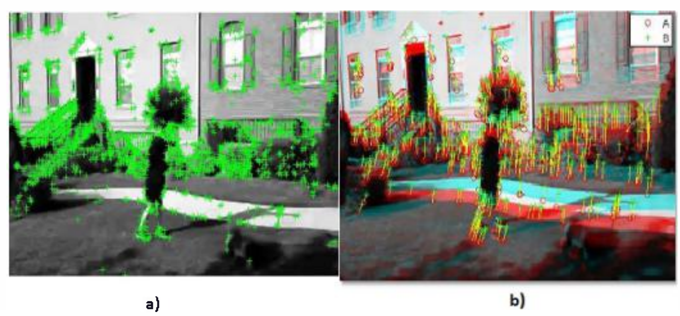

Fig. 3. (a) Corners detected by FAST algorithm (b) Corresponding features between frame A and frame B. Features of frame A are marked green and features of frame B are marked red

B. Training Features

Once we obtain a stabilized video we extract features from each of these frames. We have extracted the following two features for training the classifier:

1) Yellow Color Plane Intensity Feature: The color of a tennis ball is yellowish-green. An object of yellow color in yellow color plane appears white. We use this color cue to segment the ball in the frame [13].
   In this step, indexing was done to extract the three color planes from the image. We create a matrix that represents intensity of yellow using Eq. 4 , with a threshold to separate the background from the ball. Figure 4 gives an example of yellow plane extraction.

$$Y = g - \frac{r}{1.45} - \frac{b}{1.45} \tag{4}$$

Here, Y represents yellow color plane and r, g, b indicates the red, green and blue planes respectively. As depicted in the figure 4, the yellow tennis ball in a full color video frame appears white in yellow color plane. When thresholding is applied to it, with experimentally obtained value of greater than 11, the ball is extracted as it appears as a bright object.

a) b) c)

Fig. 4. (a) Sample Frame (b) Frame as appears in yellow color plane (c) Frame after thresholding is applied

2) Phase Quaternion Fourier Transform (PQFT) Feature: Saliency features are the candidates of attention for human eyes in an image. We use phase spectrum of Fourier transform to obtain the location of salient areas. In the video frames, as computed in [14] each pixel of the image is represented as a quaternion-
   a) RG color channel
   b) BY color channel
   c) Intensity channel
   d) Motion channel

   Then its Phase Fourier Transform is computed, subsequently a saliency map is developed. PQFT provides effective information of the salient features in an image and is a state-of-art in saliency detection. Figure 5 shows the Phase Quaternion Fourier Transform saliency map. It is evident that the most salient object to the human eye is the ball. So on computing the saliency map of the frame, the ball is clearly segmented. We further threshold the image with a value less than 0.5 to eliminate the unwanted background details.

a) b) c)

Fig. 5. (a) Sample Frame (b) PQFT saliency map of the frame (c) Frame after thresholding is applied

C. Random Forest Classification

Random Forests (RF) is an ensemble classifier that consists of many decision trees and the output of the random forest classification is the class which is mode of the outputs of the individual decision trees. Random forests were introduced in machine learning community by Breiman [15].This classification algorithm has been widely used in Computer vision for various supervised classification tasks. Advantages of RF over other machine learning classifiers include (i) computational efficiency (ii) handling of multiple features (color,texture,shape,depth etc) (iii) combines the output of multiple randomized decision trees into single classifier [16]. In RF [15] multiple decision trees are grown independently and can be grown in parallel. We have trained 100 decision trees on existing data. While training a tree, each node of the tree has access to only a randomly chosen subset of the entire set of features with replacement. At each internal node, the feature is selected from the randomly chosen feature subset which best segments the sample image. For example for a classification problem with x features, $\sqrt{x}$ features are used in each split of the node. In our case, we have used two features and the default node splitting criteria i.e. square root of number of features. During testing phase, the random forest classifies each pixel of the test image and the predicts that class to which that pixel belongs i.e. 0 or 1 by taking majority vote of the decision trees.

D. Blob Analysis

The segmented image frames resulting from random forest classification is put to blob analysis as few frames have multiple blobs similar to the ball blob. The eccentricity of all the blobs is calculated and the blob with eccentricity equal to 1 is selected as it appears as the most circular object in a video frame.

E. Video Annotation

The bounding box position of the ball blob is obtained and the ball candidates in the corresponding input video frames are annotated.

III. Experimental Results and Discussions

In this section, we first introduce data sets used in our evaluation. We then discuss the performance metrics.

A. Data Sets for Evaluation

We train our classifier on Achanta's dataset [17] using 100 images resized to 300*300 and use the corresponding binary masks as labels. The features are computed in MATLAB 2014a and random forest segmentation of the frames is performed in Python. Time taken to obtain segmented ball candidates in a frame of size 500*500 is 1.76 sec/frame.

B. Video Stabilization Results

The code for video stabilization successfully computes a stabilized frame in approximately 1 second on a 2.3 GHz Intel Dual core i5 processor. The programming platform used is MATLAB 2014. Tests were executed on a number of sample shaky video sequences with diverse contents and it was observed that the framework efficiently stabilizes the video footage. The mean of 10 raw frames and 10 stabilized frames from a sample video taken from YouTube is shown below and the mean images of the original videos are more blurred than that of the stabilized video. This stabilization technique is capable of addressing the stabilization issues typical of a video captured from a camera mounted on a UAV or a quadcopter. Figure 6 shows the results obtained by video stabilization. Figure 7 shows the results obtained by blob analysis and final annotated ball results. For ball trajectory generation, we plot the distance of ball candidates from the top-left corner of the first frame against the sequence number of the video frame. Figure 8 shows an example of such a plot which illustrates the location of all ball candidates over time. If a moving object is successfully detected in each frame, it will be depicted by a smooth trajectory over a (relatively) long period of time. However, a non ball object may exhibit short trajectories or no trajectory at all. By visualizing the plot in figure 8, it is evident that the ball candidates are correctly detected in all the frames

Fig. 6. (a) Mean of 10 original input frames (b) Mean of 10 corrected frames

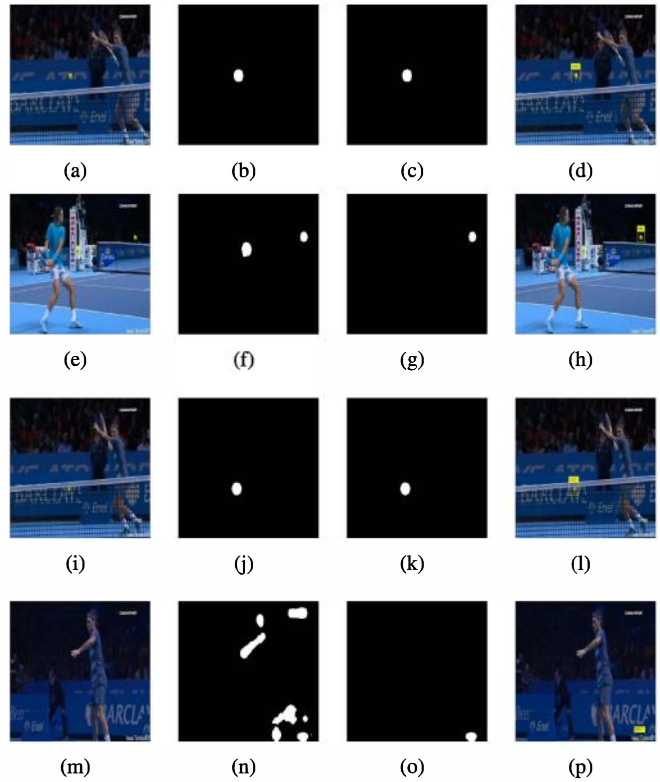

Fig. 7. Results of ball detection. Figure 7(a),7(e),7(i),7(m): Input Frame; 7(b),7(f),7(j),7(n): Segmented frame; 7(c),7(g),7(k),7(o): After blob analysis; 7(d),7(h),7(l),7(p): Annotated frames depicting a case when the ball is successfully detected in a frame, case when the ball is successfully detected in a frame containing more than one blobs after random forest segmentation, case of occlusion, case of false ball candidate detection in the absence of a true ball candidate in the frame respectively.

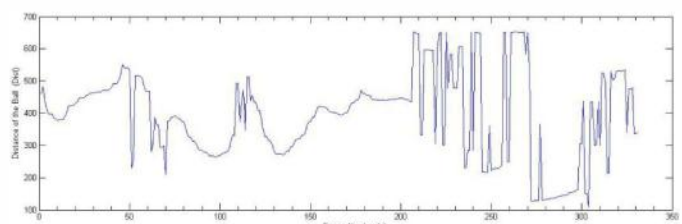

Fig. 8. Figure depicts the ball trajectory obtained in a test video sequence. n is the serial number of the frame in the sequence and d is the distance between the ball candidate from the top left corner of the first frame.

C. *Performance Evaluation*

The proposed algorithm has been evaluated on video sequences of tennis shots played by Roger Federer obtained from Youtube. As shown in Table I, the algorithm has achieved 94.75% accuracy in video 1, video 2, video 3 respectively. We compared the results obtained by our method to the statistics given in the work by Yu et al [5] (Table II). The authors [5] have used three ideas simultaneously to successfully track a ball in a tennis match video-tracking by ball candidate detection, tracking by trajectory generation and tracking by computing the missing location. However, we have achieved better results in tracking the ball, solely by using a novel ball candidate detection approach. For the best tracked video sequence we achieve 94% accuracy in comparison to 85% accuracy (best) by Yu et al obtained for ball detection.

IV. Conclusion

The current work proposes a standalone algorithm for video stabilization and tennis ball tracking using a combined computer vision and machine learning based approach.The algorithm incorporates video stabilization techniques for stabilizing the shaky video followed by a random forest segmentation approach for extracting ball candidates. Robust features like yellow color plane intensity and phase quaternion Fourier transform have been used in random forest segmentation. Results are better than the current state of the art methods. Also, our method is successful in detecting the ball even when it is partially occluded by the net

TABLE I
PERFORMANCE EVALUATION USING RANDOM FORESTS FOR THE BALL DETECTIONS

Video Sequence No.	Duration (in Sec)	Total No. of Frames	No. of Frames With Ball Candidates Available (X)	No. of Frames True Ball Candidates Detected (Y)	Accuracy(Y/X)
1	11	332	237	223	94%
2	10	302	266	200	75%
3	13	390	360	172	47%

TABLE II
COMPARATIVE PERFORMANCE ANALYSIS FOR THE METHODS

Method	Total No. of Frames	No. of Frames With Ball Candidates Available (X)	No. of Frames True Ball Candidates Detected (Y)	Accuracy(Y/X)
Yu et al [5]	341	294	250	85%
Our Method	341	237	223	94%

on the tennis ground or the racket. The model can be extended to a real time version and include player tracking and shot classification modules. Other features like shape of the ball can be added to the feature pool during random forest segmentation.

REFERENCES

[1] G. S. Pingali, Y. Jean, and I. Carlbom, "Real time tracking for enhanced tennis broadcasts," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on.* IEEE, 1998, pp. 260–265.

[2] B. Ekinci and M. Gokmen, "A ball tracking system for offline tennis videos," in *International Conference on Visualization, Imaging and Simulation*, 2008.

[3] H. Miyamori and S.-i. Iisaku, "Video annotation for content-based retrieval using human behavior analysis and domain knowledge," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on.* IEEE, 2000, pp. 320–325.

[4] K. Teachabarikiti, T. H. Chalidabhongse, and A. Thammano, "Players tracking and ball detection for an automatic tennis video annotation," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on.* IEEE, 2010, pp. 2461–2494.

[5] X. Yu, C.-H. Sim, J. R. Wang, and L. F. Cheong, "A trajectory-based ball detection and tracking algorithm in broadcast tennis video," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 2. IEEE, 2004, pp. 1049–1052.

[6] I. Kolonias, J. Kittler, W. Christmas, and F. Yan, "Improving the accuracy of automatic tennis video annotation by high level grammar," in *Image Analysis and Processing Workshops, 2007. ICIAPW 2007. 14th International Conference on.* IEEE, 2007, pp. 154–159.

[7] G. Pingali, A. Opalach, and Y. Jean, "Ball tracking and virtual replays for innovative tennis broadcasts," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4. IEEE, 2000, pp. 152–156.

[8] N. Owens, C. Harris, and C. Stennett, "Hawk-eye tennis system," in *IEE conference publication.* Institution of Electrical Engineers, 2003, pp. 182–185.

[9] F. Yan, W. Christmas, and J. Kittler, "A tennis ball tracking algorithm for automatic annotation of tennis match," in *British machine vision conference*, vol. 2, 2005, pp. 619–628.

[10] V. Lepetit, A. Shahrokni, and P. Fua, "Robust data association for online application," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–281.

[11] Y. Wang, R. Chang, T. W. Chua, K. Leman, and N. T. Pham, "Video stabilization based on high degree b-spline smoothing," in *Pattern Recognition (ICPR), 2012 21st International Conference on.* IEEE, 2012, pp. 3152–3155.

[12] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[13] S. W. Foo, "Design and develop of an automated tennis ball collector and launcher robot for both able-bodied and wheelchair tennis players-ball recognition systems." Ph.D. dissertation, UTAR, 2012.

[14] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Computer vision and pattern recognition, 2008. cvpr 2008. ieee conference on.* IEEE, 2008, pp. 1–8.

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests." in *BMVC*, 2008, pp. 1–10.

[17] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on.* IEEE, 2009, pp. 1597–1604.