The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Real Time Tennis Match Tracking with Low Cost Equipment

Mihai Polceanu,* Andreea-Oana Petac,*
Hassan Ben Lebsir,* Bruno Fiter, Cédric Buche ***
* LAB-STICC, ENIB, France, ** ERICSSON, France
[polceanu,petac,h3benleb,buche]@enib.fr, bruno.fiter@ericsson.com

Abstract

We describe a modular system able to track players and ball rebound locations during a tennis match in real time, using a single, fixed medium resolution camera. The main challenge consists in obtaining a set of efficient detectors for use with low-cost equipment to perform object and event tracking in a tennis match. Each algorithm is described and system accuracy and module execution times are evaluated.

Introduction

Match analysis is an important means for tennis players to improve their skills, whether they play professionally or recreationally. Analytics can be performed during the match, such as correct posture, ball rebound (bounce) location or mobility, or after the match has finished via statistical player data. In the case of professional matches, where referees are in charge of observing the game, conflicts may occur due to differences of opinion on rebound locations (in or out), while amateur matches may not be refereed at all. To address these issues, computer vision systems have been developed to perform tracking, conflict resolution and match analysis. During a typical tennis match, the ball is shot – sometimes at velocities exceeding $200km/h$ – by each of the players onto their opponent's court. Solo training matches are also common, where players attempt to improve their skills on particular zones of the court. The main functionality of a tracking system is to detect the ball and its trajectory as well as the players from video input. In order to evaluate line calls or to provide feedback on the zone that was hit, such a system is also required to estimate the coordinates of the tennis court in the video. While complex systems exist for automatic tracking and analysis in high-end tennis championships, we focus on solving this problem with inexpensive equipment that is easily accessible for amateur and recreational players. Hardware costs, execution speed and easy calibration are important. Raspberry Pi boards and associated camera modules are low cost hardware widely available and easy to use, which we use for video capture in our approach.

Related Work

There are several systems that are already out on the market that resemble the one we describe in this work. While the topic of tennis match analysis has long been of interest (Pingali, Jean, and Carlbom 1998; Pingali, Opalach, and Jean 2000), one of the most notable systems that is in direct relation with our work is Hawk-Eye (Owens, Harris, and Stennett 2003) – a complex computer system that is officially used in numerous sports including tennis, in order to visually track the trajectory of the ball, perform line calls and display a record of its statistically most likely path in virtual reality reconstructions. Hawk-Eye uses an ensemble of up to ten high-speed cameras placed at various locations around the tennis court which require careful calibration. The costs of installing this system (rough estimate of more than 60.000 USD) are far beyond the material capabilities of medium or lower ranked tennis championships. Since, a number of approaches have been proposed to solve the challenges posed by tracking players and tennis balls. Ball tracking was discussed in terms of both single lower resolution camera (Yan, Christmas, and Kittler 2005) and multiple cameras (Conaire et al. 2009; Renò et al. 2016), most of which perform background subtraction to limit the computational costs of the system (Mao, Mould, and Subramanian 2007). A particularly useful strategy is to compute the trajectory over a time window of possible ball candidates (Zhou et al. 2015). Player tracking is achieved in a similar way to ball detection, using only regions of the video stream that change and therefore contain motion (Teachabarikiti, Chalidabhongse, and Thammano 2010; Archana and Geetha 2015), but color-based segmentation has also been proposed (Jiang et al. 2009). Other commercial analysis tools for tennis games exist such as Mojjo (Mojjo 2017), allowing club members to analyze and replay the matches they have played on a specially prepared field (using landmarks). This application is able to perform the ball detection, remove breaks during the game to create a montage of important moments and to prepare relevant game statistics. Being much less expensive than Hawk-Eye ($\sim$3000 USD), this system targets small clubs but can still be considered expensive for recreational amateur games. Mojjo does not offer real time feedback, the game is recorded and post-processed to create the final annotated video for use by the players. Another system called In/Out (In/Out 2018) of-

fers an affordable solution (∼200 USD) for real time match analysis to amateur players. It consists in a custom built device with two cameras which can be fastened to a tennis net post and is designed for line calls. Its custom hardware and closed source and is difficult to reproduce with widely available components. In/Out uses computer vision models used in self-driving cars to provide line and pedestrian detection, however its error margin is much larger than Hawk-Eye's.

Hardware and Image Correction

The camera used in our system is a 5 megapixel sensor for Raspberry Pi I equipped with a 170° angle fisheye lens. The price for this particular camera is currently around 30 USD. The fisheye lens is important as it allows the entire court to be in sight, but requires additional system resources to correct for radial and tangential distortions and project detections on a 2D "map" of the match. Due to radial distortion, straight lines will appear curved and the distortion will increase proportionally to the distance from the focal point in the image. This poses difficulties for the line detection used to recognize the tennis court. The second type of distortion is the tangential distortion which occurs because of the curvature of the lens; this has the effect that some areas in image may appear nearer than expected. To correct this distortion, the lens' distortion coefficients must be estimated as well as taken into account technical information about the camera, such as its intrinsic and extrinsic parameters. Intrinsic parameters include information directly related to the camera model such as focal length, optical centers, etc. – this is also called the camera matrix. It depends on the camera only, so once calculated, it can be stored for future purposes. It is expressed as a 3x3 matrix. Extrinsic parameters correspond to rotation and translation vectors which transform the coordinates of a 3D point to a coordinate system. The correction of these distortions is done by providing samples of a well defined pattern; in our case, we used a chessboard pattern. By measuring known sized squares of the chessboard in an input image we can compute its distortion coefficients and therefore correct this distortion for any given point or the entire image.

Real Time Tracking

The main constraint that we impose on our approach is that it should run in real-time on low-cost equipment *i.e.*, accessible video capture hardware and a machine with average computing resources. The system is composed of several specialized modules that perform certain tasks. Each module may require input and provide output structured data. For instance, ball detection requires that the video frame is captured and is able to provide the estimated ball location as well as bounces. An application of the system consists in the choice of a list of compatible modules which communicate between them via data messages. In the following we describe the main modules (Figure 1) that constitute the proposed system (Figure 2).

Figure 1: Output of main detection modules: (a) court detection, (b) player detection and (c) ball detection.

Court Detection

Due to the fact that court detection is only needed to calibrate the system and is done only once provided that the camera stands still (as is our case), the system can afford to spend more processing resources on this task. The module first computes a color-based segmentation of the input image, to extract the court region, assuming it is the largest object in the scene. This limits the search space and eliminates outliers. Then, color is used again to compute a segmentation of the lines under the assumption that line color is white, which is the case for all standard tennis courts; this could be changed for a custom designed court. After segmentation, we apply blur, edge detection (Canny 1986) and dilation to remove outliers and obtain smooth lines. We then use Hough transform line detection to obtain a collection of local approximations of the field (Figure 1.a). Finally, we use the detections to find corners on the field, which we later map to a 2D representation via a perspective transformation. A more detailed description of the perspective transform is given in subsection "2D and 3D Visualization". This approach to detecting the tennis court is particularly slow, as we show in the "Evaluation" section, but provides a robust approach to correct localization given various possible camera placements (*i.e.*, different orientations of the court in the video stream). Court detection is used to calibrate the system and is only performed once at startup, or in the event that the camera is moved (in our setup we used a fixed camera), and therefore does not have an impact on the overall execution speed of the system.

Player Tracking

The player detection module uses motion-based background subtraction to find the most likely locations of each player in the game on which it performs tracking. For our purposes, we considered the case of 1-vs-1 matches, *i.e.*, one player in each half of the field, although the approach can be extended to 2-vs-2 games. First, contours are computed on the image resulting from motion segmentation. Assuming players are the largest moving objects, the module selects the first largest blobs, depending on the number of players. The center of the bounding rectangles of these regions provides the location information for each player. To distinguish between players and local noise caused by changes in lighting, tracking is obtained by apply a Kalman filter on each of the frame-wise detections (Joshi and Thakore 2012). Having more than one player introduces a correspondence problem between each detection (largest blobs) and the approximated player positions (Kalman filter models). The module solves

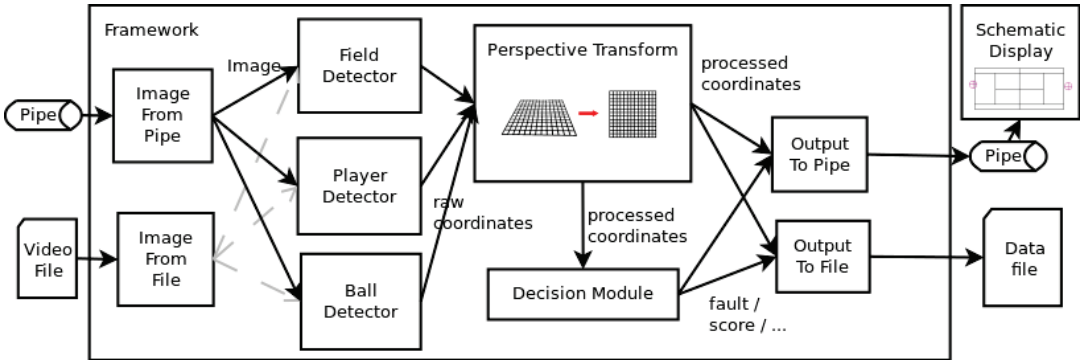

Figure 2: Overall system architecture, featuring task-specific modules and message passing between them. From online video stream (pipe) or offline files, each detection is combined into the final output for display, further analysis or saving to a log file.

this problem using a nearest-neighbor approach where new detections are associated with the nearest modeled player instance. The result of this approach is illustrated in Figure 1.b. While this method has low computational costs (see "Evaluation" section), there are situations in which it does not give accurate results due to lack of player motion while changing lighting conditions. The module alleviates this issue to some extent by using a speed limit for player location estimations so that random noise is less likely to influence predictions and, in the case of nearby noise, the tracker is able to recover more quickly.

Ball Tracking

Ball detection consists in a series of transformations applied to a sequence of motion frames. As each motion frame represents the relative motion from a given frame to the next, we merge all motion within a time window to obtain the trace of the ball by taking the maximum at each pixel of the frame. This is similar to flattening the axis of time over a short interval (we found 10 frames or 1/3 seconds given a 30fps video to be sufficient) which results in all motion being visible in a single frame. The following step involves reducing the cumulative noise caused by merging the motion frames. Motion outliers often occur due to slight changes in luminosity, but are usually sparse. Under the assumption that true motion is consistent in time, we expect that actual noise to occur but remain sparse, while real motion remain structured. To reduce noise, we apply erosion to eliminate sparse outliers and then dilation (Joshi and Thakore 2012) to recover and enhance motion traces that are likely to correspond to real movement in the video. To avoid more outliers coming from outside the field, based on results from the Court Detection module we can limit the search space to the area containing the tennis court. This reduces both computation time and false positives from spectators or light conditions. Next, we reduce the data to be processed from 2D to 1D by retaining only columns in the motion frame that contain a small number of pixels which most likely represent the ball trajectory and saving their average height to a 1D array. Note that no information about the trajectory is lost, since the height of the trajectory at each horizontal point is retained as a value in the array, while the array length corresponds to the width of the frame. To further remove outliers, two smoothing steps are applied over a small and large window respectively in which outlying values relative to the moving average are removed. In order to detect bounces, we compute local trajectory slope using linear interpolation. This finally enables us to detect changes in slope that correspond to locations of ball bounces. The final result of ball detection is shown in Figure 1.c which illustrates the trajectory, past and current rebound of the ball.

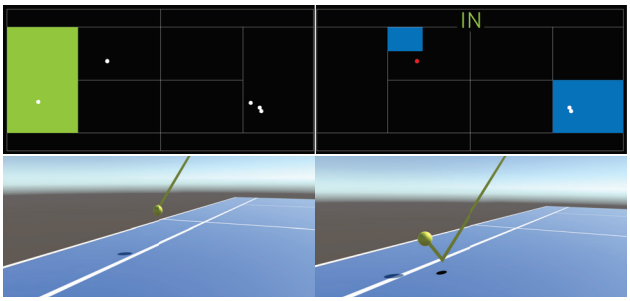

Figure 3: 2D/3D view of a tennis match. Visualization illustrates rebound locations over time, and last zone hit (top left) and custom zones defined during a training exercise (top right). 3D virtual reality reconstruction of ball rebound location (bottom). It can be used by players or referees in real time to make line calls.

2D and 3D Visualization

After applying distortion correction to account for the fisheye lens, the perspective must be canceled also in order to transform from camera view to a flat 2-dimensional map of the court. Having a 2D representation of the match is required for event analysis and statistics during the game. The perspective transformation matrix is computed using four corners of the tennis court found by the field detection module and the corresponding locations on the known 2D map.

Because this type of transformation does not account for distorted images, the detections are first corrected for lens distortion and then used to compute the perspective transform. We note that performing detection on distorted images

and only correcting the results gives a very important advantage in terms of overall execution speed of our approach. The resulting 2D view is illustrated in Figure 3 (top) in which the zone in the last rebound occurred is highlighted.

The system can also be used for training, by defining zones on the field where the player has to send the ball (Figure 3 top right). Zones of various size can be designated at different locations for which the system detects whether the shots were correctly made and provides this information to the player. Using the location of a rebound and the estimated direction of the ball, the system can provide a 3D virtual reality replay of the shot (Figure 3 bottom) to better visualize the point of impact for line calls.

Evaluation

Tests on multiple videos of recorded tennis matches showed that our system is able to detect, on average, 9 out of 10 rebounds correctly (approximately 90% accuracy). We investigated the cases in which failure occurs and noted that due to the average camera quality, the ball is not visible at all the times during shots. Currently, the ball detection module performs local prediction of the ball trajectory and requires improvements to obtain better results. We evaluated the execution speed of the main modules (Table 1) on a mainstream laptop computer. The video was provided by the Raspberry Pi camera module at a resolution of 1280×720 and a rate of 25 frames per second. Evaluation showed that the system is able to process the video at a faster rate than that at which it was captured and can also alleviate delays due to network streaming.

Motion	Ball	Players	Court
10.2 ms	7.2 ms	5.3 ms	54.6 ms

Table 1: Average execution time in milliseconds for background extraction, ball, player and court tracking on a 1280×720 video.

The system was tested live during a professional tennis championship, where a sizable screen was installed to display the detection output as well as the 2D representation for spectators, while the players were in a match. The Raspberry Pi camera module was fixed on a wall on the side of the tennis court, and the video was streamed through the network to a laptop computer for processing, which displayed the results on the screen.

Conclusions and Future Work

In this work we proposed a set of detectors for the real time analysis of a tennis match, consisting in ball trajectory and rebound detection, player tracking and court detection for automatic calibration. We described each task-specific algorithm and integrated them in an extensible, modular framework. Our focus was on affordable hardware that can be easily acquired by the large public and evaluated our approach in terms of execution time and tracking performance. We obtained a fast system that is able to provide information to players in real time with reasonable accuracy, with equipment available to the majority of amateur and recreational tennis players. The main line for improvement is increasing rebound detection accuracy. This can be obtained through a data-driven approach by using machine learning models to classify whether portions of the ball trajectory contain a rebound. As the problem itself is not highly nonlinear and relatively low-dimensional, we expect to obtain better results with this method in the future.

References

Archana, M., and Geetha, M. K. 2015. Object detection and tracking based on trajectory in broadcast tennis video. *Procedia Computer Science* 58:225–232.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* (6):679–698.

Conaire, C. Ó.; Kelly, P.; Connaghan, D.; and O'Connor, N. E. 2009. Tennissense: A platform for extracting semantic information from multi-camera tennis data. In *16th International Conference on Digital Signal Processing*, 1–6.

In/Out. 2018. http://www.inout.tennis [02/2018].

Jiang, Y.-C.; Lai, K.-T.; Hsieh, C.-H.; and Lai, M.-F. 2009. Player detection and tracking in broadcast tennis video. *Advances in Image and Video Technology* 759–770.

Joshi, K. A., and Thakore, D. G. 2012. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering* 2(3):44–48.

Mao, J.; Mould, D.; and Subramanian, S. 2007. Background subtraction for realtime tracking of a tennis ball. In *VISAPP*, 427–434.

Mojjo. 2017. https://www.mojjo.fr [02/2018].

Owens, N.; Harris, C.; and Stennett, C. 2003. Hawk-eye tennis system. In *International Conference on Visual Information Engineering*, 182–185. IET.

Pingali, G. S.; Jean, Y.; and Carlbom, I. 1998. Real time tracking for enhanced tennis broadcasts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 260–265.

Pingali, G.; Opalach, A.; and Jean, Y. 2000. Ball tracking and virtual replays for innovative tennis broadcasts. In *15th International Conference on Pattern Recognition*, volume 4, 152–156. IEEE.

Renò, V.; Mosca, N.; Nitti, M.; Guaragnella, C.; D'Orazio, T.; and Stella, E. 2016. Real-time tracking of a tennis ball by combining 3d data and domain knowledge. In *TISHW*, 1–7.

Teachabarikiti, K.; Chalidabhongse, T. H.; and Thammano, A. 2010. Players tracking and ball detection for an automatic tennis video annotation. In *ICARCV'10*, 2461–2494.

Yan, F.; Christmas, W.; and Kittler, J. 2005. A tennis ball tracking algorithm for automatic annotation of tennis match. In *British machine vision conference*, volume 2, 619–628.

Zhou, X.; Xie, L.; Huang, Q.; Cox, S. J.; and Zhang, Y. 2015. Tennis ball tracking using a two-layered data association approach. *IEEE Transactions on Multimedia* 17(2):145–156.