BALL EVENT RECOGNITION USING HMM FOR AUTOMATIC TENNIS ANNOTATION

I. Almajai, J. Kittler, T. de Campos, W. Christmas, F. Yan, D. Windridge and A. Khan

CVSSP, University of Surrey, Guildford GU2 7XH, UK

ABSTRACT

A key element for video indexing and summarisation is the description of isolated events and actions. In the context of many sports the motion of the ball plays an essential role in describing events. Due to the difficulty of ball tracking, specially in standard broadcast video, this cue has been overlooked by most researchers, in particular for games of tennis, in which the ball resolution is very small and it moves very fast. A data association method has reported a high level of success on tennis ball tracking, but so far this tracker's output has only been processed by a method based on manually crafted rules for event recognition. This set of rules use cues such as proximity between ball and players or court lines. We present an HMM paradigm to automatically learn to identify events from ball trajectories and demonstrate that its ability to capture the dynamics of the ball movement lead to a much higher performance.

Index Terms— HMM, event detection, sports annotation

1. INTRODUCTION

The ability to identify events from the trajectory of moving objects is relevant for a broad range of applications, including traffic analysis [1], surveillance [2] and sports [3]. We concentrate on the latter domain because it is based on a well-structured set of rules.

The literature related to the application of HMM for sports is quite rich for the problems of shot (or context) classification [4] and at high level analysis of the games syntax. However, the same is not observed for individual event detection within play shots.

Among the few works in this line is that of Petkovic *et al.* who use HMMs in order to classify different types of strokes in tennis games (e.g. forehand, backhand and serve) [5]. They use Fourier descriptors for the posture of the segmented players. Kijak *et al.* also use an HMM to classify a tennis game into these scenes: first missed serve, rally, replay, and commercial break [6]. They use global visual features including dominant colours, spatial coherency and camera motion activity combined with audio cues. In the context of baseball games, global visual and motion features are also used in [7] with a discrete HMM to detect four general events: base hit, strikeout, ground outs, and air outs.

None of theses pieces of work actually use the ball's motion for event detection. One exception is the work of Rea *et al.* [8] in the context of snooker games, in which tracking the white ball is a far easier problem.

Our aim is to build an automatic annotation system that is able to process video from a standard TV broadcast signal in PAL or NTSC. The description of tennis matches is dominated by ball events, but tracking the ball is a major challenge which has inhibited researchers from using the ball trajectory for event detection. To the best of our knowledge, the data association method of [9] offers the best solution to this problem. In [3], the output of the above tracker is converted into sequence of events to then automatically reason about the evolution of tennis matches. An HMM-based architecture is used to recognise high-level events such as the award of a point. However, at an intermediate level, i.e., to detect serves, hits and ball bounces, [3] uses a number of rules manually crafted specifically for tennis. These rules relate the ball position at the instant of a velocity change with the position of other objects, such as court lines and players. External entities are also used. For instance, a shape analysis method evaluates the outline contour of the player to detect the serve action.

Further to being able to automatically annotate tennis games, we aim to design a system that can easily be generalised for other court sports, such as badminton. Therefore, it is not viable to use crafted heuristics in order to detect events. For this reason we propose the use of HMM to detect such events by analysing the ball motion alone, without having to rely on external methods, such as a velocity change detector and a player action recogniser.

Our experiments show that the proposed HMM approach is not only more generalisable (as the event detector is learned, rather than manually designed), but it is also more robust. We obtained better results than the system in [3] on the same dataset.

A global picture of our tennis annotation system is described in Section 2, which also gives further details of the event detection method described in [3]. A description of the proposed HMM-based method is provided in Section 3. Section 4 describes our experiments and Section 5 concludes this paper.

Thanks to EPSRC for funding through grant EP/F069626/1 - ACASVA.

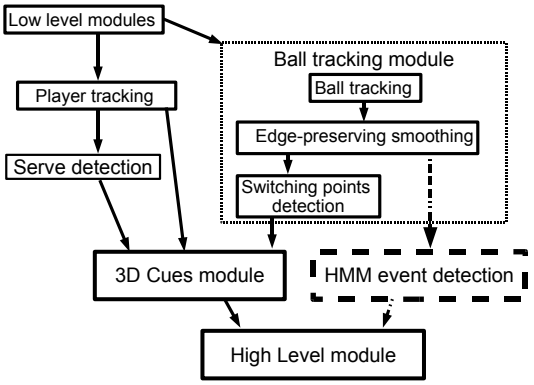


Fig. 1. Key modules of the tennis annotation system. The two arrows leading to the 'high level module' indicate the two different approaches contrasted in this paper.

2. THE TENNIS ANNOTATION SYSTEM

Figure 1 gives an overview of the modules of the tennis annotation system indicating where the present method differs from the method described in [3]. In Section 2.1, we briefly describe the low and high level modules which are used both in [3] and in the method of the present paper. The event detection method ('3D Cues') of [3] is reviewed in Section 2.2.

2.1. Common modules

Processing single view videos as they are streamed for TV (PAL/NTSC) is very challenging because they include camera motion, replays, close-ups of players and commercial breaks. To deal with these problems, we follow the low-level methods described in [3].

For ball tracking (the 'ball tracking module' in Fig. 1), a special background subtraction process is done in order to classify foreground blobs as ball / not ball using an SVM. Features consist of gradient direction at blob boundary, colour and size [9].

The ball tracks are established in two stages. First, "tracklets" are built from sets of strong object candidates in the form of 2nd-order (roughly parabolic) trajectories. These correspond to intervals when the ball is in free flight. Then a graph-theoretic data-association technique is used to link the tracklets into complete ball tracks [9].

The next subsection gives an overview of the event detection method reported in [9, 3]. In section 3, we describe the proposed alternative. Table 1 lists 17 tennis event labels used in the annotation system, with their description. The output of the event detection method is a sequence of events of each play shot. This sequence is processed by a discrete HMM that models the awarding of points in tennis (the 'high level modlue'). The award of games and sets in the match is achieved by grammars that reflect the rules of the tennis game.

Table 1. Summary of tennis events used [3]

Event	Description
SFR	Serve by Far player, Right Side
SFL	Serve by Far player, Left Side
SNR	Serve by Near player, Right Side
SNL	Serve by Near player, Left Side
BIF	Bounce Inside Far player's half court
BOF	Bounce Outside Far player's half court
BIN	Bounce Inside Near player's half court
BON	Bounce Outside Near player's half court
HF	Hit by Far player
HN	Hit by Near player
BIFSR	Bounce Inside Far player's Serve area on the Right
BIFSL	Bounce Inside Far player's Serve area on the Left
BOFS	Bounce Out of Far player's Serve area
BINSR	Bounce Inside Near player's Serve area on the Right
BINSL	Bounce Inside Near player's Serve area on the Left
BONS	Bounce Out of Near player's Serve area
NET	Bounce on NET

2.2. Tennis-specific heuristics for event detection

Sudden changes in the velocity of the ball are recorded as "ball event candidates". An algorithm of generalized edge-preserving signal smoothing [10] is used to detect these key events (model switching points) and interpolate ball positions in the frames where the ball is not detected.

An example of the final tracking and event detection result is shown in Fig. 2. In this example there are no false positive events; however 3 events (1 bounce, 2 hits) are not detected. In [3], the points at which the ball changes its motion abruptly

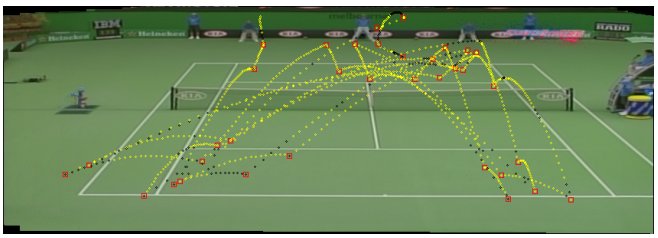


Fig. 2. Ball tracking result with detected event candidates. Yellow dots: detected ball positions. Grey dots: interpolated tennis ball positions. Red squares: detected key events.

are taken as key events such as hit and bounce. For serve detection, the system performs these three operations on each player: (i) verify if any of the players is located in a possible serving position and create a contour of each player, (ii) analyse the shape of the outline of each player to classify it as a serve hit, (iii) verify that the tennis ball is directly above the player. The process is repeated for each frame from the beginning of the shot, and terminates if, for *any* of the players *all* of the above holds. At this point a serve is deemed to have occurred.

For every identified key event (switching point), the position of the players is used to help distinguish between bounces

and hits. If a player is near the ball, the event is more likely to be a hit. Established bounces are checked against the court lines to see if they are in or out on either side of the court. The net event is detected when the ball is in the region around the court net. The events above are reinterpreted in 3D space using a homography computed for each play shot.

3. HMM EVENT DETECTION

The edge-preserving algorithm used to detect the model switching points gets false positives and false negatives. Under detection, giving rise to false negatives occur when an interpolated edge is not sharp enough to be considered a key event. The serve detection method might also fail in detecting a serve and that can lead to key events highlighted by the ball tracker to be ignored or confused with hit events. Moreover, knowing where events might have happened will still leave uncertainty about its type (bounce/hit etc). All these mistakes can accumulate to an extent that the high level interpretation module may not absorb them, leading to interpretation errors.

We explore the alternative of using a set of continuous-density left-to-right first-order HMMs, Λ, to analyse the ball trajectories and detect K events.

$$\Lambda = [\lambda_1, ..., \lambda_k, ..., \lambda_K] \tag{1}$$

Given enough ball trajectory data including velocity and acceleration, such HMMs are expected to model each event type. Observations, $\mathbf{o}_t$, are thus composed of ball coordinates and their derivatives.

$$\mathbf{o_t} = [x_t, y_t, x_t^{'}, y_t^{'}, x_t^{''}, y_t^{''}] \tag{2}$$

Each HMM used is characterised by three probability measures, namely, the state transition probability distribution matrix (A), the observation probability distribution (B) and the initial state distribution (π), defined for a set of N states $S = (s_1, s_2, ..., s_N)$, and ball information observation sequence $\mathbf{O} = \mathbf{o}_1, ..., \mathbf{o}_T$. The probability $b_j(\mathbf{o}_t)$ of generating observation $\mathbf{o}_t$ at state j and time t, is given by

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M_j} c_{jm} N(\mathbf{o}_t, \mu_{jm}, \mathbf{\Sigma}_{jm}) \tag{3}$$

where M_j is the number of mixture components in state j, c_{jm} is the weight for the m^{th} component and $N(\mathbf{o}_t, \mu_{jm}, \mathbf{\Sigma}_{jm})$ is a multivariate Gaussian with mean μ_{jm} and diagonal-covariance $\mathbf{\Sigma}_{jm}$.

After evenly dividing the evidence among the models, embedded training is used to learn the models parameters. It allows model boundaries (event boundaries) to shift through a probabilistic entry into the initial states of each model [11]. This training is an iterative process that seeks to maximize the probability that the HMMs account for the training sequences. Once HMMs are trained, the most likely state sequence for a new observation sequence can be calculated using the Viterbi algorithm [11]. The inferred state sequence provides the decoded sequence of events.

4. EXPERIMENTS

The dataset is composed of manually annotated play shots of two different matches of the 2003 Australian Open tennis championship (53 play shots of Women's final and 48 play shots of Men's final). The annotation does not include event boundaries. Based on the events listed in Table 1, a set of 17 HMMs with five emitting states per model are trained using the HTK toolkit [11]. To calculate the observations velocity and acceleration, it is found empirically that the performance is maximised when the size of the velocity window is six and the size of the acceleration window is two. Training and testing is accomplished by a leave-one-out process because of the small size of the data set. Following the training process described in section 3, the HMMs are re-estimated using embedded Baum-Welch re-estimation. Every time the number of mixture components in every state is increased, one cycle of embedded re-estimation is applied. Viterbi decoding is used with the following grammar:

$E1 {$E2}, where

$E1 = <SFR BINSL> | <SFL BINSR> | <SNR BIFSL> | <SNL BIFSR> | <SFR NET> | <SFL NET> | <SNR NET> | <SNL NET> | <SFR BONS> | <SFL BONS> | <SNR BOFS> | <SNL BOFS>;

and

$E2 = <HF BIN> | <HN BIF> | <HF BON> | <HN BOF> | <HN HF> | <HF HN> | <HF NET> | <HN NET> | HN | HF | BIF | BIN | BON | BOF.

This type of grammar is explained in [11]. It basically forces the decoded sequence here to start with a serve event, followed by bounce in or out the serve area or net events. Then a variable number of rally hits and bounces is expected before the tennis play shot is over.

To analyse the accuracy of the ball event detection by the proposed and the previous method, the module output is matched with the correct reference transcriptions. This comparison is performed using dynamic programming to align the two transcriptions and then count substitution, deletion and insertion errors. Figure 3 shows the gain in accuracy as the number of mixture components increases with static features only, static features with velocity and static features with velocity and acceleration. The inclusion of velocity features significantly increases the event detection accuracy and an extra gain can be achieved when acceleration features are included. An accuracy of 84.3% event detection was achieved with 128 Guassian per state compared to 69.65% of the previous method. The low detection accuracy of the previous method is due to the lack of a good hit/bounce distinction mechanism and higher rates of false positives and false nega-

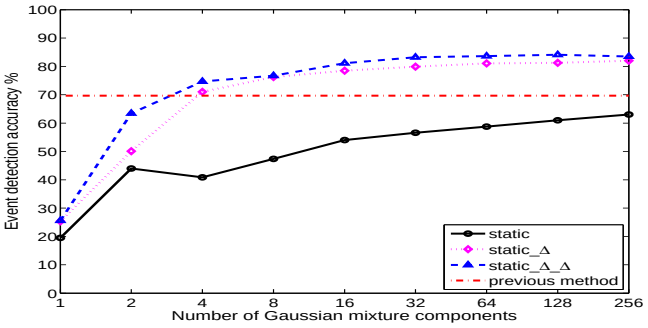
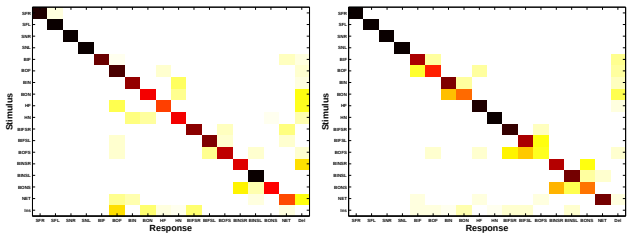


Fig. 3. Event detection accuracy at different Guassian mixtures.



(a) previous method (b) proposed method

Fig. 4. Confusion matrix of the two event detection methods. The rows and columns are indexed following the order shown in Table 1.

tives. A comparison the two detection methods in terms of confusion matrices is shown in figure 4. The last column of the matrix represents the number of deletions (false negatives) and the last row represents the number of insertions (false positives). It can be seen that the off-diagonal areas of the proposed method have less intensity compared to the previous method, exhibiting lower false positives and false negatives rates. It also shows the proposed method to be much better in hit/bounce distinction but with slightly more confusion between bounce in and out events especially when the ball is close to the line. This confusion is expected to diminish with more training data for the HMM to learn the court areas.

5. CONCLUSION

We proposed a HMM-based method to detect events from ball trajectories in tennis games. This method was compared with a system that uses a manually crafted set of heuristics which use the detected velocity changes of the ball as key points and describe the events according to the proximity between the ball and court lines and players. The players shape is also taken into account in that system. Our approach provides a simpler and more effective framework which sidesteps the need for external modules to describe events. One of our plans for future work is to investigate if the accuracy of the proposed method can be increased by providing training data where events are manually labeled with time boundaries rather than just the event type.

6. REFERENCES

[1] Xiaokun Li and Fatih M. Porikli, "A hidden markov model framework for traffic event detection using video features," in *ICIP*, 2004.

[2] P. Remagnino and G. A. Jones, "Classifying surveillance events from attributes and behaviour," in *BMVC*, 2001, pp. 685–694.

[3] I. Kolonias, *Cognitive Vision Systems for Video Understanding and Retrieval*, Ph.D. thesis, University of Surrey, 2007.

[4] Z. Xiong, R. Radhakrishnan, A. Divakaran, Y. Rui, and T. S. Huang, *A Unified Framework for Video Summarization Browsing and Retrieval with Applications to Consumer and Surveillance Video*, Elsevier Academic Press, 2006.

[5] M. Petkovic, W. Jonker, and Z. Zivkovic, "Recognizing strokes in tennis videos using Hidden Markov Models," in *Intl. Conf. on Visualization, Imaging and Image Processing*, 2001.

[6] E. Kijak, G. Gravier, P. Gros, L. Oisel, and F. Bimbot, "HMM based structuring of tennis videos using visual and audio cues," in *ICME*, 2003.

[7] C.C. Lien, C.L. Chiang, and C.H. Lee, "Scene-based event detection for baseball videos," *Journal of Visual Communication and Image Representation*, vol. 18, no. 1, pp. 1–14, 2007.

[8] N. Rea, R. Dahyot, and A. Kokaram, "Modeling high-level structure in sports with motion-driven HMMs," in *ICASSP*, 2004.

[9] F. Yan, A. Kostin, W. Christmas, and J. Kittler, "A novel data association algorithm for object tracking in clutter with application to tennis video analysis," in *CVPR*, 2006, vol. 1, pp. 634–641.

[10] V. Mottl, A. Kostin, and I. Muchnik, "Generalized edge-preserving smoothing for signal analysis," in *IEEE Workshop on Nonlinear Signal and Image Analysis*, 1997.

[11] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.0*, Cambridge University Press, 2000.