A TWO LAYERED DATA ASSOCIATION APPROACH FOR BALL TRACKING

Xiangzeng Zhou[1], *Qiang Huang*[2], *Lei Xie*[1], *Stephen Cox*[2]

[1]Shaanxi Provincial Key Laboratory of Speech & Image Information Processing (SAIIP), School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, P. R. China
[2]School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, United Kingdom

ABSTRACT

Ball-tracking is a key technology in processing and analyzing a ball game. Because of the complexity of visual scenes, a large number of objects are usually selected as candidates for the ball, leading to incorrect identification, and conversely, the true position of the ball may sometimes be missed. In this paper, we propose a two layered data association method to improve the robustness of ball-tracking. At a local layer, we use a sliding window based Token Transfer method to generate a set of sub-trajectory candidates. At a global layer, a single ball trajectory is obtained by applying a dynamic programming based splice method to a graph consisting of the sub-trajectory candidates. We evaluated our approach on tennis matches from the Australian Open and the U.S. Open, and the results obtained show that our approach outperforms the state-of-art approach by around 30 %.

Index Terms— data association, tennis, ball tracking, trajectory, layered

1. INTRODUCTION

Sports video analysis is currently receiving increasing attention. It has a number of useful and beneficial applications, such as highlight extraction [1], tactics analysis [2], computer-assisted refereeing [3], etc. Robust detection and tracking of figures and objects in the game is the foundation for high level analysis. Knowledge of the position of the ball at any time is also essential for more ambitious systems that attempt to "understand" a game [4].

This paper is concerned with ball tracking, which has been traditionally approached as a data association task using a Markov chain assumption and a predictive motion model, applied frame by frame. However, data association in candidate "clutter" is difficult because of false positives (non-ball objects detected as balls) and false negatives (undetected balls). After probabilistic data association (PDA) was first proposed by Bar-Shalom and Fortmann [5], many researchers have strived to improve the robustness of this task [6] [5] [7]. Viterbi Data Association (VDA) [7] uses a parallel search scheme using the Viterbi algorithm. Robust data association (RDA, [6]) treats data association as a motion fitting problem in an attempt to provide robustness to abrupt motion change (e.g. when the ball is struck). Yan *et al.*'s work [8] describes a hierarchical scheme with a graph-theoretic formulation that attempts to overcome some of RDA's limitations (e.g. the lack of motion smoothness constraints). More recently, Huang *et al*. [9] described a Viterbi-based method to estimate ball trajectory: however, it loses some precision in tracking the target, mainly because of the lack of a well-defined motion model.

In this paper, we describe a two layered approach for tennis ball tracking that attempts to increase robustness against both abrupt motion change and the absence of the tracked object. Given a set of ball candidates, at the "local" layer we use a Token Transfer method to generate a set of sub-trajectories as the input to the next layer. At the "global" layer, we use a dynamic programming based splice scheme to obtain the optimal sub-trajectory combination as the final trajectory. The paper is organized as follow. Ball candidate extraction and the local method are described in Section 2. A description of the global layer approach is given in Section 3. Evaluation results are shown in Section 4, and conclusions are presented in Section 5.

2. LOCAL LAYER: N-BEST SUB-TRAJECTORIES EXTRACTION

2.1. Candidates Extraction

To extract ball candidates in each frame, we difference two adjacent frames and choose candidates using size and color filters. Two kinds of mask are then used to discard false candidates. Firstly, we extract the court lines using the direct linear transformation (DLT, [10]) to discard those candidates that occur on the court lines. Secondly, we apply a mean shift based method [11] to track players and discard ball candidates that occur inside players' regions. We denote the set of ball candidates extracted in frame f_t as $\mathbf{C}_t = \{C_t^1, C_t^2 \ldots, C_t^{N_t}\}$ $(t = 1 \ldots T)$.

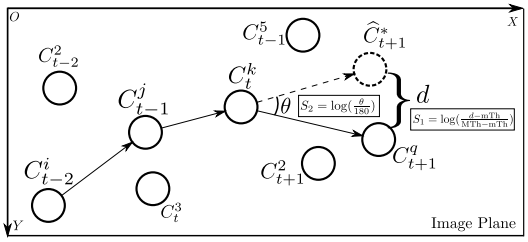


Fig. 1. Prediction of ball position using a linear acceleration model and detected ball candidates. Filled circles represent actual detected ball candidate positions and the dotted circle a predicted position.

2.2. Motion Model and Quality Score

Next, we define a directed network using the following link rule: each candidate can link forwards to any candidate located in its neighborhood $[mTh, MTh]$ in the next frame. MTh and mTh represent the maximum and minimum pixel distances a tennis ball can travel in $\triangle T$ (the reciprocal of the frame rate): these values are here set experimentally to 65 and 5 respectively. The nodes of the resulting network are termed Candidate Nodes (CNs), and each path through the network corresponds to a sub-trajectory candidate.

To predict the position of a ball in frame f_{t+1}, we "score" each candidate C^q_{t+1} using a local motion model estimated using the positions of previous candidates in three previous frames, and the predicted candidate $\widehat{C}^*_{t+1}$ given by the motion model. The local motion model assumes constant acceleration. Velocity V^k_t and acceleration Acc^k_t are estimated using equations 1 and 2 below, and are used to predict a candidate position $\widehat{C}^*_{t+1}$ in equation (3).

$$Acc^k_t = \frac{(C^k_t - C^j_{t-1}) - (C^j_{t-1} - C^i_{t-2})}{\triangle T^2} \tag{1}$$

$$V^k_t = \frac{C^k_t - C^j_{t-1}}{\triangle T} + Acc^k_t \times \triangle T \tag{2}$$

$$\widehat{C}^*_{t+1} = C^k_t + V^k_t \times \triangle T + \frac{Acc^k_t \times (\triangle T)^2}{2} \tag{3}$$

Figure 1 depicts a typical prediction and score for a candidate. We define two "quality" scores for C^q_{t+1} in frame f_{t+1}:

$$S_1 = \log(\frac{d - \mathrm{mTh}}{\mathrm{MTh} - \mathrm{mTh}}) \tag{4}$$

$$S_2 = \log(\frac{\theta}{180}) \tag{5}$$

where d and θ are the pixel distance and the angle between $\widehat{C}^*_{t+1}$ and C^q_{t+1}, as shown in Fig. 1. Then the quality score for any particular path in the directed network is $S_1 + S_2$ for all triplets inside the path. To avoid large negative scores for S_1 and S_2, we constrain their range to the interval $[S_m, 0]$ where S_m is predefined.

2.3. Token Transfer

To search for the optimal path, we propose a modified version of the Viterbi algorithm which we call *Token Transfer*. A *Token* holds two quantities, of which *CNPath* is the history of CNs that the Token has passed through, and *Score* is the accumulated score (i.e. the sum of S_1 and S_2 over all triples in *CNPath*).

The longer a *CNPath* becomes, the more chance there is that it incorporates false ball candidates. We utilize a windowing technique to alleviate this problem. A *Token Transfer* process, shown in **Algorithm 1**, is executed afresh within each window (in our work, the window length and sliding step length are set to 21 and 5 empirically). Here, $\mathbf{Toks}^i_t$ represents a set of Tokens passed up to Candidate Node C^i_t ($\in \mathbf{C}_t$); tok_0 is an initial Token whose *Score*=0, *CNPath*=$\emptyset$. In window w_n, we retain the *N* top-ranked Tokens for each frame (we used $N = 2$). The *CNPath* of these selected Tokens thus comprise a sub-trajectory set, $\mathbf{Toks}_{w_n}$, and are used as the input to the global layer.

Algorithm 1 Token Transfer

Initialization:
$\mathbf{Toks}^i_1 = \{\mathrm{tok}_0\}$;
Recursion:
for all f_t within each window w_n **do**
  if $\mathbf{Toks}^i_t == \emptyset$ **then**
    $\mathbf{Toks}^i_t = \{\mathrm{tok}_0\}$;
  end if
  $\mathbf{Toks}^j_{t+1} = \emptyset$;
  for all $tok \in \mathbf{Toks}^i_t$ **do**
    for all $C^j_{t+1} \in \mathbf{C}_{t+1}$ **do**
      Calculate S_1 and S_2;(S_1=0, S_2=0 when $t < 3$)
      $tok\rightarrow$Score = $tok\rightarrow$Score $+S_1 + S_2$;
      $tok\rightarrow$CNPath = $tok\rightarrow$CNPath $+C^j_{t+1}$;
      Add tok into $\mathbf{Toks}^j_{t+1}$;
    end for
  end for
  Pick the *N* top-ranked Tokens from $\mathbf{Toks}^j_{t+1}$ into $\mathbf{Toks}_{w_n}$;
end for

3. GLOBAL LAYER: SUB-TRAJECTORIES SPLICE

At the global layer, our goal is to determine an optimal subset from the set of sub-trajectories $\mathbf{Toks}_{w_n}$, to form the final ball trajectory. To do this, we firstly construct a directed acyclic graph (DAG) based on these sub-trajectories, and then search for the optimal directed path through the DAG using Dynamic Programming.

3.1. Construction of Directed Acyclic Graph

We illustrate the kind of sub-trajectories we obtain in Fig. 2(a). Here, each column represents a window (w_n) and contains a small number of sub-trajectories ($Tok^i_{w_n}$). These sub-trajectories, also called Token Nodes (TNs), are then used to form a grid structure as shown in Fig. 2(b). The DAG is constructed by linking TNs. Three different cases

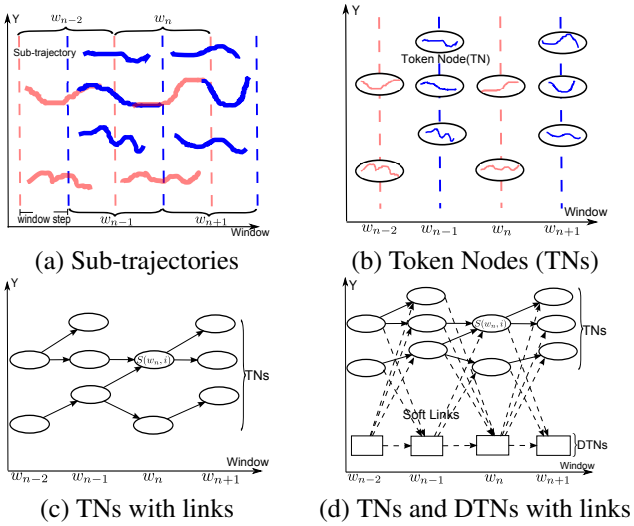
are possible, depending on the relationship between the two TNs ($Tok^i_{w_n}$ and $Tok^j_{w_{n+1}}$) occurring in adjacent windows w_n and w_{n+1}:

Case1. Temporal and spatial overlap:
If the front portion of *CNPath* of $Tok^j_{w_{n+1}}$ is the same as the rear portion of the *CNPath* of $Tok^i_{w_n}$), the two TNs are connected with a directed arc from $Tok^i_{w_n}$ to $Tok^j_{w_{n+1}}$.

Case2. Temporal overlap only:
There is no directed arc between $Tok^i_{w_n}$ and $Tok^j_{w_{n+1}}$.

Case3. No temporal overlap:
For this case, we linearly extend $Tok^i_{w_n}$ at its end point (in frame f_{t_1}) and linearly extend $Tok^j_{w_{n+1}}$ at its start point (in frame f_{t_2}) along the direction of the velocity vectors of the two points. If a junction of the two extended lines exists and the distance between the two end points is less than $MTh \times (t_2 - t_1)$, we connect the two TNs with a directed arc.

After connecting all tokens, we obtain a DAG, as shown in Fig. 2(c), in which the score of a TN in the nth window is defined as $S(w_n, i)$. However, not every window contains true sub-trajectories: these windows should be skipped. Hence, shown in Fig. 2(d), for each window we append a Dummy Token Node (DTN) whose *CNPath*=$\emptyset$ and *Score*=0. DTNs can connect to any TN and DTN, which we term Soft Link.



(a) Sub-trajectories (b) Token Nodes (TNs)

(c) TNs with links (d) TNs and DTNs with links

Fig. 2. Construction of DAG from (a) to (d) step by step.

3.2. Sub-Trajectories Splice

To form the optimal trajectory, we employ a Dynamic Programming approach to process the obtained DAG. The full algorithm is presented in **Algorithm 2**. We define two recursive quantities $F(w_n, i)$ and $G(w_n)$. As illustrated in Fig. 3, $F(w_n, i)$, computed only on TNs, indicates the minimum cost of the paths ending up at the ith TN in w_n. $G(w_n)$, computed only on DTNs, represents a set of costs of paths ending up at

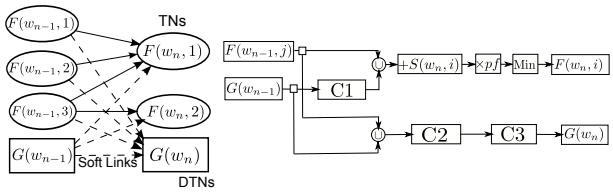


Fig. 3. Diagrammatic description of Algorithm 2.

the DTN in w_n. In addition, $P(w_n, i)$ stores the backtracking information of the optimal path. Figure 3 gives a diagrammatic explanation of how to compute $F(w_n, i)$ and $G(w_n)$ recursively, and the algorithm is shown in **Algorithm 2**.

In order to prevent a path from using only the dummy tokens when traversing the DAG, we set three conditions:

***C*1:** a path consisting only of dummy tokens is removed
***C*2:** a path containing three consecutive dummy tokens at the end is removed
***C*3:** the two top-ranking paths are retained in the recursion

We also assume that trajectories in which the direction changes very quickly are unlikely to be correct. Hence when searching through the DAG, we use a penalty factor (*pf*) defined as:

$$pf = \frac{LCount}{\text{Length of Partial Path}} - 1 \tag{6}$$

LCount is found by examining the angle between each pair of consecutive path segments in each candidate path and noting the number of angles whose value is larger than a threshold (set to 70 degrees here). During searching for the optimal path, we weight $F(w_n, i)$ by pf—hence the smaller the value of pf is, the more likely the path will be selected.

4. EVALUATION

4.1. Experimental Setting

In our experiments, we used material from two tennis matches. One was a men's singles match from the 2010 Australian Open and the other a men's singles match from the 2011 US Open. We extracted 37206 and 24363 frames from the two match videos at a sampling rate of 25 frames per second. The ground truth was obtained by manually locating and storing the ball's position in a frame. Frames in which the ball did not appear were labeled as inactive frames. After application of the candidate extraction schemes described in Section 2.1, the average number of detected ball candidates per frame was 4.5.

The metric used for evaluating algorithms is the $F1$-score:

$$\text{F1} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall}) \tag{7}$$

$$\text{Precision} = n_{tp}/(n_{tp} + n_{fp}) \tag{8}$$

$$\text{Recall} = n_{tp}/(n_{tp} + n_{fn}) \tag{9}$$

where:

Algorithm 2 Sub-Trajectories Splice

Initialization:

$G(w_1) = \emptyset,\ F(w_1, i) = S(w_1, i);$

Recursion:

1: **for all** $n = \{1, 2, ..., N\}$ **do**
2: $\mathcal{G} \triangleq \{G(w_{n-1}) \mid C1\};$
3: $F(w_n, i) = \min_j\{\{F(w_{n-1}, j) \cup \mathcal{G}\} + S(w_n, i)\} \times pf;$
4: $G(w_n) = \{F(w_{n-1}, j) \cup G(w_{n-1}) \mid C2 \wedge C3\};$
5: $P(w_n, i) = \arg\min_j\{F(w_n, i) \cup G(w_n)\};$
6: **end for**

Termination:

7: $F^* = \min_i\{F(w_N, i) \cup G(w_N)\};$
8: $P^* = \arg\min_i\{F(w_N, i) \cup G(w_N)\};$

$n_{tp} \triangleq$ # frames where a ball was detected and was present,
$n_{fp} \triangleq$ #frames where a ball was detected but wasn't present,
$n_{tn} \triangleq$ # frames where a ball wasn't detected and wasn't present,
$n_{fn} \triangleq$ #frames where a ball wasn't detected but was present.

In our experiments, the values of n_{tp}, n_{fp}, n_{tn} and n_{fn} were obtained by comparing the distance between the actual ball position and the estimated ball position in each frame. We treat the hypothesised candidates as false candidates if this distance is larger than 10 pixels.

4.2. Evaluation

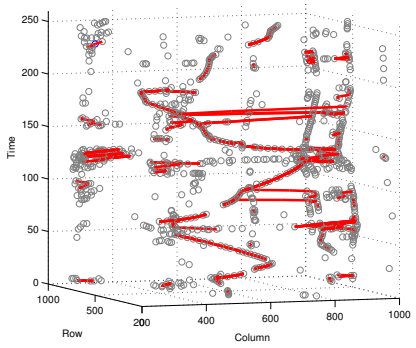
Figure 4 illustrates an example of ball tracking using our approach. Figure 4(a) plots the sub-trajectory set obtained in the local layer, while Fig. 4(b) shows the final estimated trajectory.

Table 1. The performance of three editions of method

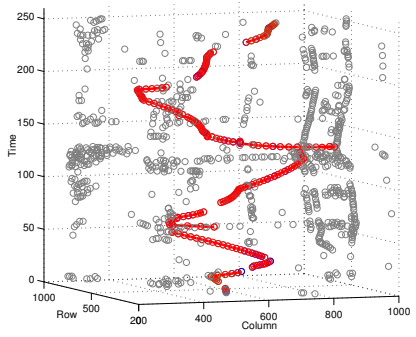
Data	Method	Precision (%)	Recall (%)	F1 (%)	TP Err. (pixels)
Aust.	Huang[9]	44.17	50.49	47.12	1.83
	TL	72.01	72.50	72.26	1.94
	TL+PF	73.13	73.11	73.12	1.94
U.S.	Huang[9]	73.33	60.69	66.41	1.53
	TL	81.47	67.27	73.69	1.97
	TL+PF	82.31	67.52	74.19	1.79

Table.1 summarizes the performance of our two-layered (TL) approach compared with a baseline, which is the results using the method introduced in our previous work [9]. The baseline method uses the Viterbi algorithm to search a globally smooth trajectory amongst weighted candidates, but there is no motion model and the Viterbi search is less sophisticated.

It is clear that, when the $F1$ score is considered, our proposed approach outperforms the baseline by a significant margin. However, the pixel error (TP Err.) is slightly higher in all cases when compared with the baseline. This suggests that the new approach is much better at identifying the presence or absence of a ball in a frame, but the estimates of the ball's position are not quite as good as the baseline technique. In most cases of interest, this slight loss of accuracy in position information is more than compensated for by the higher accuracy in identifying that the ball is present or absent from the frame. Note that the addition of the smoothness penalty factor (PF) always improves the $F1$ score, and for the U.S. match, lowers the pixel error considerably.



(a) The local layer



(b) The global layer

Fig. 4. A complete example of our approach.

5. CONCLUSION AND FURTHER WORK

In this paper, we have presented a two layered data association method for tennis ball tracking in a complex scene. Experiments show that our approach is significantly more robust to false detections than the technique used in our previous work. In the future, we will extend the approach to integrate audio information. This will lay the foundations for making a high level analysis of game and making an analysis of player's actions and tactics.

6. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (61175018), the Natural Science Basic Research Plan of Shaanxi Province (2011JM8009), the Fok Ying Tung Education Foundation (131059) and a grant from the UK Engineering and Physical Sciences Research Council (EP/F069626/1).

References

[1] Ying Yang, Shouxun Lin, Yongdong Zhang, and Sheng Tang, "Highlights extraction in soccer videos based on goal-mouth detection," in *Proc. 9th Int. Symposium on Signal Processing and Its Applications*, 2007, pp. 1–4.

[2] Guangyu Zhu, Qingming Huang, Changsheng Xu, Yong Rui, Shuqiang Jiang, Wen Gao, and Hongxun Yao, "Trajectory based event tactics analysis in broadcast sports video," in *Proc. 15th Int. conf. on Multimedia*, 2007, pp. 58–67.

[3] Roke Manor Research Ltd, "Hawk-eye," 2003, http://www.bbc.co.uk/pressoffice/pressreleases/stories/2003/06_june/10/hawk_eye.shtml.

[4] Qiang Huang and S. Cox, "Inferring the structure of a tennis game using audio information," *IEEE Trains. on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1925 – 1937, sept. 2011.

[5] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[6] Vincent Lepetit, Ali Shahrokni, and Pascal Fua, "Robust data association for online applications," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 281–288.

[7] T. Quach and M. Farooq., "Maximum likelihood track formation with the viterbi algorithm," in *Proc. IEEE Conf. on Decision and Control*, 1994, pp. 271–276.

[8] Fei Yan, Alexey Kostin, William Christmas, and Josef Kittler, "A novel data association algorithm for object tracking in clutter with application to tennis video analysis," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006, pp. 634–641.

[9] Qiang Huang, Stephen Cox, Xiangzeng Zhou, and Lei Xie, "Detection of ball hits in a tennis game usingaudio and visual information," in *Proc. APSIPA Annual Summit and Conference*, 2012.

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[11] D Comaniciu, V Ramesh, and P Meer, "Kernel-based object tracking," *IEEE Trains. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 564–577, 2003.