

DETECTOR-LESS BALL LOCALIZATION USING CONTEXT AND MOTION FLOW ANALYSIS

Fabio Poiesi, Fahad Daniyal, Andrea Cavallaro

Queen Mary University of London
Mile End Road, E1 4NS, London, United Kingdom
Email: {fabio.poiesi, fahad.daniyal, andrea.cavallaro}@elec.qmul.ac.uk

ABSTRACT

We present a technique for estimating the location of the ball during a basketball game without using a detector. The technique is based on the analysis of the dynamics in the scene and allows us to overcome the challenges due to frequent occlusions of the ball and its similarity in appearance with the background. Based on the assumption that the ball is the point of focus of the game and that the motion flow of the players is dependent on its position during attack actions, the most probable candidates for the ball location are extracted from each frame. These candidates are then validated over time using a Kalman filter. Experimental results on a real basketball dataset show that the location of the ball can be estimated with an average accuracy of 82%.

1. INTRODUCTION

Sport analysis has recently gained much attention for automatic highlights generation, camera selection and video summarization. In games that involve a ball such as basketball or soccer, the focus of attention is primarily on the ball and players around it. Although player tracking has already been addressed as a feature-based multi-target detection problem, ball detection and tracking is still a challenging task due to its low visibility and abrupt motion.

Ball detection methods generally estimate the position of the ball using *spatial features*. Moreover, several approaches perform an additional *temporal smoothing* to filter out incorrect estimates. Methods based on spatial features generally perform an exhaustive search for a set of features such as color, shape and size. To estimate candidate locations of the ball, Miura *et al.* [1] use predetermined ranges of size and color histograms. Assuming the motion of the ball is linear, a binary map is generated where all “true” values correspond to potential candidate regions. This approach is further modified with object segmentation in [2]. They segmented the goal and the ellipse to improve ball size estimation. D’Orazio *et al.* [3] use the Circular Hough Transform (CHT) to locate circular objects within an image. The CHT is used to detect candidate regions that are then validated by a neural classifier. Miura *et al.* [1] demonstrated that elements of the environment such as lines, stands and players with white uniforms can cause false candidates. Furthermore, assumptions such as fixed size and circular shape of the ball do not hold anymore when the ball moves at a high speed, which makes it appear elongated. Moreover, approaches based on spatial features tends to fail in the presence of other circular objects such as the character “O” on advertising or certain circular logos.

This work was supported in part by the EU, under the FP7 project APIDIS (ICT-216023).

Methods based on verifying the temporal consistency of candidate detections use various forms of tracking. Liang *et al.* [4] extract the ball candidates from several consecutive frames (using color, shape and size cues and template matching) and then build a weighted graph with each node representing a candidate ball position. The Viterbi algorithm is finally employed to extract the optimal path. As an alternative, Kalman filter has been used to filter a set of disjoint ball trajectories and remove false candidates while estimating a global trajectory for the ball [2]. Choi *et al.* [5], use a first order dynamic model for the ball motion perturbed by Gaussian random noise, where the estimated trajectory of the ball is generated using a Particle filter. Treptow *et al.* [6] described the ball candidate as a state vector, whose elements are position, velocity, and size. Their dynamic model was implemented as a simple random walk with “almost” constant velocity. These methods are still dependent upon the initial detection phase, which is based on the extraction of visual features of the ball that are not reliable because of the frequent occlusions and the similarity with the background.

Unlike existing approaches, instead of using visual features associated to the ball, we estimate the ball candidates based on the location of the players and their motion during attack actions. We propose an approach for ball localization that uses contextual information to estimate the approximate location of the ball based on the players’ behavior. We use expected dynamics of the game and motion flow to estimate the regions of convergence for players and map these regions to the probable ball locations. Temporal consistency is then validated using Kalman filtering. We test the proposed approach on a real basketball scenario, where the ball is most of the time either partially or completely occluded, and compare it with alternative approaches.

This paper is organized as follow. Section 2 presents the proposed approach. Results and evaluation of the method are presented in Sec. 3. Conclusions are finally drawn in Sec. 4.

2. PROPOSED APPROACH

The estimation of the ball location using motion flow analysis is divided in two steps. First, candidate regions are estimated by evaluating the convergence points of the motion flow. The position of the flow is used to skip the non-interesting regions. Next, the candidates are temporally validated in order to obtain a consistent estimation of the ball location over time. The block diagram of the proposed approach is shown in Fig. 1.

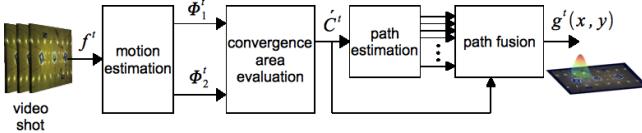


Fig. 1. Block diagram of the proposed approach.

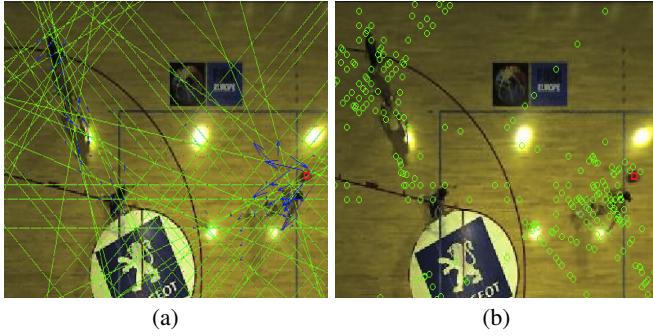


Fig. 2. An example of finding the ball candidates: (a) straight lines (green) drawn on the motion vectors (blue) and (b) intersection lines. The position of the ball is shown with a red circle.

2.1. Generation of candidate regions

In order to obtain the candidate regions for the ball position we analyze the points of convergence of the flow evaluated on the areas with motion flow. To extract the motion flow caused by player movement, we calculate the motion vectors for two consecutive frames, f^t and f^{t+1} . We use two different block sizes for block matching, i.e. 5×5 and 10×10 pixel. The smaller block size allows detailed localization of the flow, but it is sensitive to noise. The larger block size gives more coarse information about the direction of the flow, but with reduced sensitivity to noise. Denote the two vector fields, $\vec{\Phi}_1^t = \{\vec{\phi}_{1,1}^t, \dots, \vec{\phi}_{1,K_1}^t\}$ and $\vec{\Phi}_2^t = \{\vec{\phi}_{2,1}^t, \dots, \vec{\phi}_{2,K_2}^t\}$, which are associated to the smaller and larger block size, respectively. With $\vec{\Phi}_2^t$ we evaluate the converging areas of the flow. Since the larger block estimation is less noisy the information about the direction of the flow is more consistent. Each motion vector, $\vec{\phi}_{2,i}^t = (u_{2,i}^t, v_{2,i}^t)$ at a certain position $(x_{2,i}^t, y_{2,i}^t)$, gives the direction of the flow, we trace for each of them a straight line. The points defining the straight line are calculated as:

$$\begin{cases} x_1 = x_{2,i}^t \\ y_1 = y_{2,i}^t \\ x_2 = x_1 + u_{2,i}^t \\ y_2 = y_1 + v_{2,i}^t \end{cases}, \quad (1)$$

for all $i = 1, \dots, K_2$. Figure. 2 (a) shows a sample frame with this motion vector extension. We consider the solutions (intersections) generated from this vector extension by Gaussian elimination. The solutions we take into account are those belonging to the domain that start from the tail of the motion vector and expand to its tip (see Fig. 3). These solutions (see Fig. 2 (b)) provide us with the convergence points of the flow. Since the points of convergence are spread around the court, we are only interested in the solutions localized in the areas where there is the presence of flow (e.g., the solutions in the second half of the court are removed because there are no players that can cause flow). To this end we merge the 2D histograms of

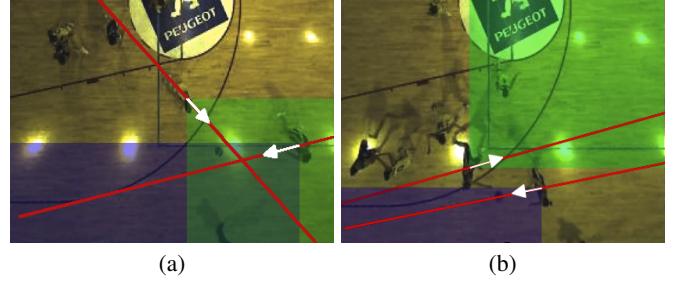


Fig. 3. Example of solution validities: (a) a solution contained in the intersection of the two domains is a valid solution, (b) a non-valid solution. Different colors of the domains belong to different vectors.

Algorithm 1 Path Estimation

```

 $\dot{C}^t$  : set of candidates for ball location at time  $t$ ;
 $\dot{c}_j^t$  :  $j^{th}$  candidate for ball location at time  $t$ ;
 $p_{\dot{c}_j^t}^t$  : probability of the  $j^{th}$  candidate to contain the ball;
 $x_j^t$  : location of the  $j^{th}$  candidate along  $x$  direction ;
 $d(x_{j_1}^t, x_{j_2}^t)$  : Euclidian distance between  $\dot{c}_{j_1}^t$  and  $\dot{c}_{j_2}^t$ ;
 $J$  : total number of candidates;
 $\mathcal{L}_j^t$  : set of candidates with  $x_j^{t-1} < x_i^t \forall i = 1, \dots, J$ ;
 $\mathcal{S}_j^t$  : set of candidates with  $x_j^{t-1} > x_i^t \forall i = 1, \dots, J$ ;
 $\mathcal{E}_j^t$  : set of candidates with  $x_j^{t-1} = x_i^t \forall i = 1, \dots, J$ ;
 $\dot{c}_B^t$  : best candidate at time  $t$ ;
 $|\cdot|$  : cardinality of a set;

1: for  $j = 1$  to  $J$  do
2:   if  $|\mathcal{E}_j^t| \geq |\mathcal{L}_j^t|$  and  $|\mathcal{E}_j^t| \geq |\mathcal{S}_j^t|$  then
3:      $\dot{c}_B^t = \dot{c}_j^t$ 
4:   else if  $|\mathcal{S}_j^t| = |\mathcal{L}_j^t|$  then
5:      $P_L = \sum_{\forall i \in |\mathcal{L}_j^t|} p_{\dot{c}_i}^t$ 
6:      $P_S = \sum_{\forall i \in |\mathcal{S}_j^t|} p_{\dot{c}_i}^t$ 
7:      $\dot{c}_B^t = \operatorname{argmin}_i d(x_j^{t-1}, \max(P_L, P_S))$ 
8:   end if
9:   if  $|\mathcal{L}_j^t| > |\mathcal{S}_j^t|$  and  $|\mathcal{L}_j^t| > |\mathcal{E}_j^t|$  then
10:     $\dot{c}_B^t = \operatorname{argmin}_i d(x_j^{t-1}, \mathcal{L}_j^t)$ 
11:   end if
12:   if  $|\mathcal{S}_j^t| > |\mathcal{L}_j^t|$  and  $|\mathcal{S}_j^t| > |\mathcal{E}_j^t|$  then
13:     $\dot{c}_B^t = \operatorname{argmin}_i d(x_j^{t-1}, \mathcal{S}_j^t)$ 
14:   end if
15: end for

```

flow positions, given by the vector field $\vec{\Phi}_1^t$ (shown in Fig. 4 (a)), and the solutions calculated with the straight lines (Fig. 2 (b)). The bin size is set to 20×20 pixels in order to contain the ball. Each bin has a value between 0 and 1 and the merging of the two histograms gives the percentage of the flow converging in that bin. Thus each bin can be regarded as a candidate for the ball position.

Let the set of these candidate ball positions at time instant t , be represented as $C^t = \{c_1^t, c_2^t, \dots, c_J^t\}$, where J is the total number of candidates. Each candidate c_j^t , $j = 1, \dots, J$ has a probability $p_{c_j^t}$ to contain the ball, such that $\sum_{j=1}^J p_{c_j^t} = 1$.

2.2. Temporal validation of the candidates

Since the candidate regions are spread around the court and might not be consistent over time, we evaluate all possible paths by choosing the most reliable entrusting on the candidate regions.

At each time instant t , the list of candidates (C^t) is sorted, and

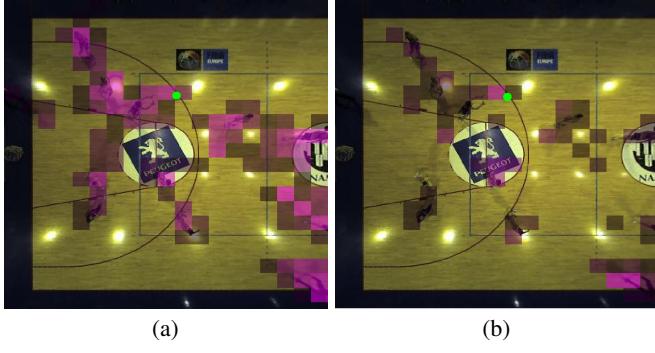


Fig. 4. Visual comparison of candidate estimation using (a) motion based detection and (b) the proposed approach without Kalman filtering. The green dot indicates the ball position.

the subset of these candidates $\tilde{C}^t = \{\tilde{c}_1^t, \dots, \tilde{c}_j^t\}$ is selected such that

$$\sum_{j=1}^J p_{\tilde{c}_j^t} = \tau, \quad (2)$$

where $\tau \in [0, 1]$. The threshold τ must select the most significative candidates and at the same time remove those with low probability. An appropriate value we found empirically for their threshold is $\tau = 0.6$. After selecting the most probable candidates in the horizontal and vertical direction, to approximate the spatial density of the candidates over time we calculate the 4th order polynomial that approximates these values. Since the polynomial gives an approximation of the concentration of the flow over time, the closest candidates to the polynomial are chosen as starting points. Example of candidates for the x coordinate and one graph for y coordinate are shown in Fig. 6 (a) and Fig. 6 (b). The initial candidates are selected within the vicinity of the polynomials. The best path for each starting position is then extracted using Algorithm. 1.

Each path is approximated with a Kalman filter (KF) (see Fig. 6 (c) and Fig. 6 (d)). The state equation and the observation equation are as follows:

$$\begin{cases} s^{t+1} = Fs^t + w^t \\ z^t = Hs^t + v^t \end{cases}, \quad (3)$$

where s^t is the internal state that represents the position of the ball at the instant t , F is the state transition, and z^t is the measurement and H is the observation model. $w^t \sim \mathcal{N}(\mu_w, \Sigma_w)$ and $v^t \sim \mathcal{N}(\mu_v, \Sigma_v)$ are the process noise and observation noise, respectively. The mean and the covariance of the noise are represented by μ and Σ .

From the estimated paths (x and y coordinates), obtained with the KF, we define a 2D Gaussian function over time (as shown in Fig. 5(b)) representative of the probability distribution for the ball location, i.e.

$$g^t(x, y) = \frac{1}{\sqrt{2\pi\sigma_x^t\sigma_y^t}} \cdot e^{-\frac{(x-\mu_x^t)^2}{\sigma_x^t} - \frac{(y-\mu_y^t)^2}{\sigma_y^t}} \quad (4)$$

where μ_x^t and μ_y^t are the mean of the paths of the x and y coordinates, respectively, and σ_x^t and σ_y^t are the associated variances. Fig. 5 shows an example extracted from a real sequence.

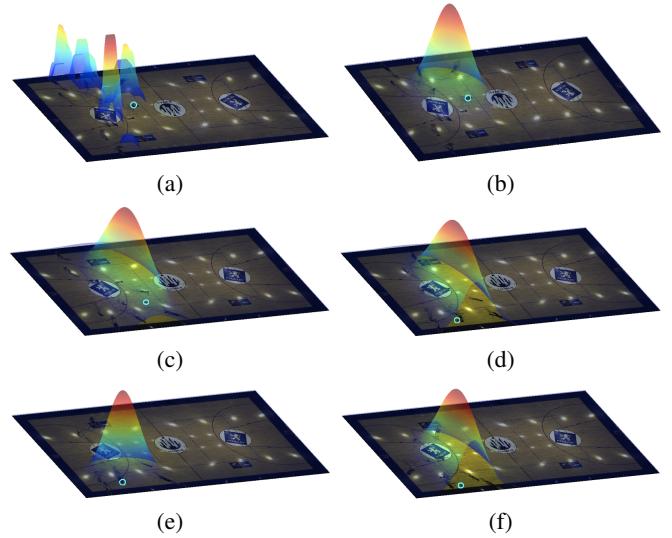


Fig. 5. (a) Candidate bins without filtering and (b,c,d,e,f) the Gaussian function after Kalman filtering.

3. RESULTS AND EVALUATION

3.1. Experimental setup

We test the proposed algorithm, flow based detection (FBD) on real basketball sequences from the APIDIS dataset¹. The video resolution is 770×430 captured at 25Hz. The experiments are performed on five sequences (see Table 1). We use top-view cameras and the flat view was obtained with image warping on the ground-plane. We extracted 865 frames out of 1892 frames of the attack phases (as described in Table 1).

We compare the proposed algorithm with a motion detection based method (MBD) and a vector extension based method (VEB). MBD (Fig. 4(a)) provides the estimation of the ball position biasing its decision as the area with the highest quantity of motion. VEB provides the estimations using the information of the flow position together with the analysis of the convergence areas (see Fig. 4 (b) and Fig. 2 (a)). The area with highest probability is selected as the candidate for ball location.

3.2. Discussion

Table 2 shows the average distance error between the groundtruth (manually extracted) and the estimate for each method. The measure of the error is given in meters (m) where $0.045m = 1$ pixel. For MBD and VEB the distance error is taken from the center of the candidate area. For the FBD the distance errors is taken from the position of the Gaussian maximum. The percentage accuracy for each method is shown in Table 2.

The distance error for the motion based detection is higher because when the players are running towards the ball the estimate is assigned in correspondence to their position and not to the point of focus. On the other hand, the error remains limited and low, with respect to the dimension of the court, because there are situations where the ball is present in close vicinity of the areas with a high quantity of motion (e.g., when a player is inside the three-point area

¹<http://www.apidis.org/Dataset/>, accessed: 8 February 2010

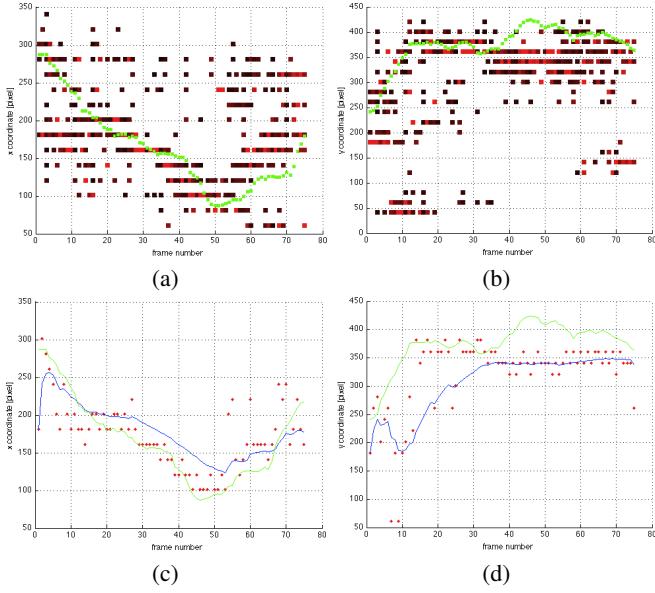


Fig. 6. An example of best path estimation using ball candidates: (a,b) horizontal and vertical coordinates of points, respectively, with groundtruth (green) and estimated positions (red). (c,d) Estimated best path using Kalman filtering.

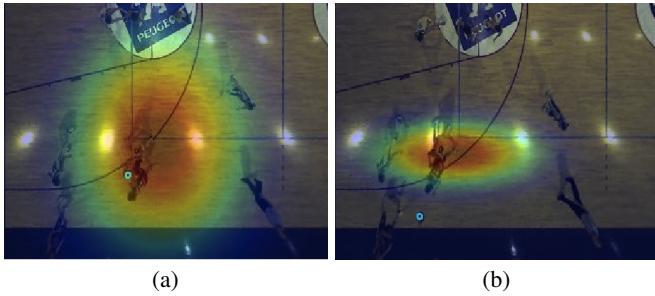


Fig. 7. Sample images with location of the ball marked for : (a) good approximation and (b) bad approximation.

trying to reach the basket). VEB demonstrated a higher accuracy compared to the MBD, but suffers a degradation in performance due to the noise generated by the non-uniform movement of the players (see Fig.2(a)). Small changes in the direction of a player, rather than the movements of the stationary players, generate wrong estimation. With FBD, the Kalman filter together with the best path selection filters this noise and provides an improvement in the estimation thus generating a lower estimation error. If this assumption fails then we have a false estimate about ball location. For instance, in Fig. 7 (a) we have a good estimate of the ball location (ball in the area of high probability) as the players are moving toward the ball, whereas in Fig. 7 (b) the ball is in the area of low probability due to a splitting of the positions of the players.

Table 1. Experimental dataset.

Seq.	Total # of frames	# of frames used
S1	514	267
S2	396	253
S3	394	162
S4	333	78
S5	255	105
Total frames	1892	865

Table 2. Performance evaluation: Average distance and the Percentage accuracy for each sequence. Key: *Motion Detection Based* (MDB), *Vector Extension Based* (VEB), *Flow based detection* (FBD).

Tech.	Sequences									
	S1		S2		S3		S4		S5	
	Err.	%	Err.	%	Err.	%	Err.	%	Err.	%
MDB	4.68	72	4.08	76	4.28	75	4.88	71	4.84	71
VEB	3.52	79	3.64	78	2.76	84	3.92	77	4.08	76
FBD	3.20	81	3.48	79	2.44	86	2.84	83	3.12	81

4. CONCLUSIONS

We have presented a novel approach for ball localization based on motion flow analysis. The proposed flow-based detection method (FBD) employs contextual knowledge about players' behavior and provides a first estimate for the location of the ball. Then Kalman filtering is used to smooth the candidate location results. The proposed approach is compared with alternative methods and achieves an average accuracy of 82% on a commonly available basketball dataset.

An extension to this work is its application to perspective-view cameras and the automatic detection of the attack phase by analyzing the trajectories of the players.

5. REFERENCES

- [1] J. Miura, T. Shimawaki, T. Sakiyama, and Y. Shirai, "Ball route estimation under heavy occlusion in broadcast soccer video," *Computer Vision and Image Understanding*, vol. 113, pp. 653–662, May 2008.
- [2] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video," in *Proc. of ACM Int. Conf. on Multimedia*, Aug. 2004, pp. 11–20.
- [3] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante, "A new algorithm for ball recognition using circle hough transform and neural classifier," *Pattern Recognition*, vol. 37, pp. 393–408, Mar. 2004.
- [4] D. Liang, Y. Liu, Q. Huang, and W. Gao, "A scheme for ball detection and tracking in broadcast soccer video," *Advanced in Multimedia Information Processing*, vol. 3767, pp. 864–875, Nov. 2005.
- [5] K. Choi and Y. Seo, "Probabilistic tracking of the soccer ball," *Statistical Methods in Video Processing*, vol. 3247, pp. 50–60, Dec. 2004.
- [6] A. Treptow and A. Zell, "Real-time tracking for soccer-robots without color information," *Robotics and Autonomous Systems*, vol. 48, pp. 41–48, Aug. 2004.