Piecewise linear prediction model for action tracking in sports

Axel Baldanza[1,2,*], Jean-François Aujol[2], Yann Traonmilin[2], and François Alary[1]

[1]Rematch
[2]Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251,F-33400 Talence, France.
[*]a.baldanza@rematch.fr

***Abstract*—Recent tracking methods in professional team sports reach very high accuracy by tracking the ball and players. However, it remains difficult for these methods to perform accurate real-time tracking in amateur acquisition conditions where the vertical position or orientation of the camera is not controlled and cameras use heterogeneous sensors. This article presents a method for tracking interesting content in an amateur sport game by analyzing player displacements. Defining optical flow of the foreground in the image as the player motions, we propose a piecewise linear supervised learning model for predicting the camera global motion needed to follow the action.**

I. Introduction

Sports tracking is an important problem in computer vision because of the increasing demand from sports institutions or television channels for automatic algorithms. Such algorithms can be used to analyze player motions, game sequences or to make cameras able to record the game automatically.

Current state-of-the-art approaches for spatial or temporal action localization use detection of objects at the frame level [1], [17], [20]. The detected objects are linked or tracked across time to process the action localization in the video. In professional sports, high-definition cameras placed above the field make ball detection possible because the ball is less occluded and more recognizable in these conditions. The information of the ball position can then be used to track the action. [9], [18], [21], [22] present adaptations of methods based on object detection at the frame level for tracking the action. These articles use players and ball detection to track the ball position. But in amateur sports, where the vertical position of the camera is variable from a game to another, and where the cameras use heterogeneous phone lenses, it is a more difficult task. For the same reasons, and because of the variability of ball appearance between different sports, other ball detection algorithms as D'Orazio et al. [2] would not be accurate in our model. We evaluated state-of-the-art neural network architectures for object detection on our test database. These architecture are used in [1], [21] to detect balls or localize action across frames. The two models tested are a YOLO (You Only Look Once) architecture [7], [15] and a faster R-CNN (Region-Based Convolutional Neural Network) [13], [16] with weights pre-trained on the COCO dataset [11]. Both models managed to detect balls in less than 5% of the frames (respectively 2.12% and 4.81%). This justifies the need for a more robust tracking method dedicated to degraded acquisition conditions. In this paper, we propose a novel supervised learning approach for tracking interest zones in sports. We define interest zone as the area of the field a cameraman would have filmed. This necessarily includes tracking the ball, but does not have to always be centered in the image to make it more stable. State-of-the-art methods for motion prediction focus on predicting people or objects motion as Fernando et al. [5] who used motion prediction for multi-people tracking. Although predicting the motion of each person in the images can be a robust way to deduce a global motion, this kind of method is hardly usable in sports videos, where the player directions have sharp and unpredictable changes. Methods based on multi-view acquisition cameras as [10], [14] can't be applied on our database because we dispose of only single view videos. To deal with these constraints, our method is based on predicting global motions of the zone of interest instead of local object motions. To predict the motion needed to track the action, we use a piecewise linear model learned on a database of amateur sport videos. Using this model, our method takes as input a sport video and returns a real number corresponding to the predicted camera motion at each frame. This motion is normalized as a percent of the total field. For this work, we used a basketball video database[1] to train and test our model. As videos are captured by human beings, we assume these videos use a ground truth tracking of the action.

The main contribution of this article is the definition of an action tracking method based on a global motion prediction while other methods focus on object detection and tracking. The first main advantage of this new method is that it allows real-time tracking as it uses only fast operators. As presented in Dosovitskiy et al. [3], optical flow can be calculated in real-time. The second main advantage of this method is that it avoids object detection which can be non robust with heterogeneous cameras and variable points of view. Finally, considering action tracking as a global motion estimation instead of localization allows us to annotate a large video database automatically, while labeling positions in videos is a time-consuming task.

[1]From Rematch platform: www.rematch.tv

II. Optical flow based approach

In this part, we describe a method based on the optical flow for deducing camera displacement. First, the method segments foreground and background. We assume that the background displacement corresponds to camera motion and we want to differentiate it from the foreground motions that we suppose to be the player motions. Warnakulasuriya et al. [19] use player motions to predict team goals. This work induces that player displacements can indicate ball and action position. Following this intuition, player motions can be analyzed to predict the position or motion of the zone of interest. The estimation of a dense optical flow [12] yields a matrix of vectors (u_i, v_i) resulting of the displacement of pixels in the image.

A. Background/foreground segmentation

Let us consider an image I as a $\mathbb{R}^N$ vector. Players are defined as the moving foreground of the image. We assume that the background occupies more than half of an image. This is justified because sports cannot be captured close enough to make players occupy more than 50% of the screen without zooming. We define the background global motion at time t as the 2D median (u^*, v^*) of the dense optical flow computed between the frames t and $t-1$

$$(u^*, v^*) = \underset{(u,v)\in\mathbb{R}^2}{\operatorname{argmin}} \sum_{i=1}^{N} ||(u, v) - (u_i, v_i)||. \quad (1)$$

with (u_i, v_i) elements of the optical flow. (u^*, v^*) is found by using iteratively reweighted least squares [6].

We define the optical flow of the foreground $f^j \in \mathbb{R}^{N\times 2}$ as the optical flow matrix composed by elements distant to a threshold θ from the principal mode (u^*, v^*) defined at equation (1) and where other elements are fixed to 0. To make the foreground values independent of the camera motion, non-zero elements of the foreground are added to (u^*, v^*). f^j is a vector of N components f_i^j defined as

$$f_i^j = \begin{cases} (u_i + u^*, v_i + v^*), & \text{if } ||(u_i, v_i) - (u^*, v^*)||^2 \geq \theta. \\ 0 & \text{else .} \end{cases} \quad (2)$$

As the threshold θ needs to be adapted for each frame, we propose to use

$$\theta = \lambda \frac{\sum_{i=1}^{N} ||(u_i, v_i) - (u^*, v^*)||^2}{N} \quad (3)$$

with λ a constant manually defined to maximise the segmentation quality.

Figure 1 shows examples of background/foreground segmentation applied with this method. In the following, we consider only the horizontal component of the foreground optical flow for simplicity of notations because basketball camera displacements used in our examples are essentially horizontal. For other sports, the vertical component can easily be added to our model.

Figure 1: On the left column, images taken from a video. In the middle, optical flow computed from the left image and the previous frame. On the right, background/foreground segmentation.

B. Environment normalization

This part presents optical flow normalization to increase the robustness of our model and to deal with the huge changes between the acquisition positions induced by the amateur conditions (position of the cameraman in the stands, distance to the field, ...). First, the lateral position of the camera is assume as known because the camera needs to be placed in the middle of the field to make our model consistent.

1) Scale: First, due to amateur conditions, the model needs to be robust to large changes in player size induced by variations of the distance between the camera position and the field. To avoid the impact of these variations, we define n_f the number of activated pixels in the foreground and n_b those activated in the background. We define $\eta \in \mathbb{R}$ as

$$\eta = \left(\frac{n_b}{n_f}\right)^{\alpha}, \quad (4)$$

where $\alpha \in \mathbb{R}$ is a parameter used to adjust the impact of scale normalization on the prediction (fixed to 1.3 in our experiments). The scale normalization is defined as above because the prediction model defined in (5) depends on the activated pixel number in the foreground. If the camera is placed far from the field, we will obtain high values for η.

2) Vertical position and camera orientation: As images are taken from a single position, we cannot normalize camera placement with a full homography. In these conditions, we use a direct linear transform [4] to normalize the camera position. Our model uses linear rescaling to normalize the impact of vertical position and camera orientation on the player appearance. As explained in the scale normalization part, our prediction model depends on the number of activated pixels in the foreground, and camera orientation can have a strong impact on player size. Depth is normalized by η. Linear rescaling is applied to the result of the segmentation f^j. To determine the rescaling for each f^j, we apply a 2 points vertical correspondence between the optical flow of the foreground and the normalized space. Figure 2 shows examples of environment normalization results on the foreground position. We see players are vertically replaced and the vertical wingspan is adjusted.

Figure 2: On the left, foreground/background segmentation in the starting space. On the right, foreground/background segmentation after linear rescaling.

C. Motion prediction

Our model aims at deducing global motion needed to track action from the player motions computed in (2). In this part, we present how our model is deducing global direction from segmented optical flow. The idea is to transform a complex action localization into a simple piecewise linear problem. The computational cost of linear prediction allows us to track the action in real-time.

We write $d^j \in \mathbb{R}$ the camera displacement we want to deduce at time j from f^j constructed as in equation (2). We can solve the problem by defining the linear model

$$d^j = \langle f^j \eta, z \rangle + e^j \tag{5}$$

where $z \in \mathbb{R}^N$ is a learned weight vector, $e^j \in \mathbb{R}$ the prediction error and η the scale normalization parameter we defined in (4).

During a video, the same player motions can induce different camera motions depending on the occurring situation. For example, all players running in a direction would require camera motion if it is located in an extremity of the field and not in the other one. According to this, our model has to be able to predict different motions depending on the situation for the same foreground optical flow.

We denote by s a situation and we extend the model (5) to the piecewise linear model

$$d^j = \langle f^j \eta, z_s \rangle + e^j \tag{6}$$

where there is a different weight vector z_s learned for each situation. In applications, the camera position is assumed known because of the calibration. The situation management can be deduced using the camera position. For evaluating the predictions on our data, situation management is done by analyzing the evolution of the ground truth values. The starting situation is annotated manually and situation changes are applied when the sum of ground truth displacements reach 100% as ground truth values are normalized in percent of the field as presented in the next part. The idea behind situations separation is to make the model adaptive to a maximum of different behavior. By defining adapted situations, this model is extendable to other sports and not only basketball.

D. Model learning

1) Labelization: In this part, we present an automatic annotation method for our database. As explained before, our model predicts a global tracking motion by linear multiplication with a weighted learned matrix. To train the model and fit weight matrices z_s defined in (6), foreground optical flows need to be labeled with associated camera motion. This is done automatically by considering the global motion needed to track action as the horizontal component u^* defined in (1) normalized as a percentage of the total field. Comparing the curves obtained with this method and video contents allow us to prove that the global computed background motion, defined as u^*, is indeed associated to the camera behavior. Assuming that the sum of these displacements is representing 100% of the field, we can normalize each motion as a percentage of the total field. This allows the model to be robust to scale changes during the training because these variations can induce different background displacements for the same camera motion.

2) Situations and training: In this work, the database is split into two situations: action starting in the left side of the field ($s = 0$) and on the right side ($s = 1$). We trained the 2 matrices z_r and z_l for each situation by using ADAM optimizer [8]. To fit the model, the loss function is defined as

$$L(z_s) = ||F_s z_s - d_s||^2 + \frac{1}{\epsilon} \sum_{i=1}^{K} \Phi_i(z_s) \tag{7}$$

where K is the batch size, F_s is a matrix composed of K normalized foreground optical flow vectors, d_s is a vector composed by the K camera motion associated labels. Φ_i are penalization functions that favor a motion direction corresponding to each situation (motion to the right for $s = 0$ and to the left for $s = 1$ typical in basketball).

$$\Phi_i(z) = max((-1)^{s+1} \langle F_s^i, z_s \rangle, 0)^2. \tag{8}$$

The learning rate is set to $\alpha = 10^{-6}$. The parameter ϵ is set to 0.1. It must be noted that the penalization Φ should be adapted to the sports considered with any prior knowledge for a given situation.

Finally, a post-processing step is performed on the predicted motion. This step is composed by a Gaussian smoothing and a thresholding to force little values to 0 and avoid the results to drift.

III. Experimental results

In this section, we study the effectiveness of our optical-flow based method to track interesting contents in basketball amateur games.

A. Dataset

The training database is made on 4050 optical flow vectors from 27 videos of 10 seconds computed of amateur videos. Videos from this database match the constraint that the camera is placed in the middle of the field to make our model consistent.

B. Tracker evaluations

Performances are evaluated on 2250 optical flows computed from 15 different videos from the same platform. Two different versions of our method are evaluated in the test section. A first version of the model with no penalization function during learning (NO PEN) and the model with the penalization functions (PEN) to justify the use of these functions. We also highlight the impact of the post processing step by giving the results of the PEN model without this post-processing (NO PP).

1) Evaluation metrics: To evaluate numerical results of the model, we use the mean absolute error (MAE) on predictions and integrated predictions (i.e. position with respect to the start of the video):

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{9}$$

where n is the number of predictions, y_i the prediction and x_i the ground truth value.

Considering time integration of prediction results highlights the gap between the predicted camera location at time t and the position of the ground truth. We consider that when the MAE on integrated predictions (IP) is over 15%, the algorithm has lost the location of the action. Analyzing MAE on standard prediction allows us to see if predictions are locally close to ground truth. In good tracking results, this value enables us to understand if the predictions are locally the same as ground truth or if the tracking is done by error compensation. This metric shows if bad results are induced by local errors or global ones.

2) Evaluation results: As there is no out of the box method designed to predict global camera motion in sport videos, comparison with state-of-the-art methods is not easy with our metric. As explained in the introduction, we tested two state-of-the-art object detection methods to track the ball on the test database and they detect balls in less than 5% of the frames. With these results, predicted camera lost match thread in all videos of test database with camera motions. Evaluation results are summarized in Table I. First, results show that post-processing improve prediction performances. The thresholding in the post-processing explains why the model is able to predict exactly the ground truth in video 11 where the camera is not moving. The two models have the same number of best following results but PEN has fewer videos where the algorithm is losing the match thread (MAE on IP >15%). This justifies the use of penalization during learning. According to the metric explained in the previous part and the perceptual analysis of the results, our method gives accurate prediction results on 14 videos of the database. Examples of accurate tracking results in Table I are detailed in Figure 3. These examples show samples where the model gives predictions close to the ground truth and where the action is precisely located. Graphs show that predicted curves are closely following ground truth. The blue rectangle in the images shows that the predicted camera is keeping the action on the screen. MAE computed on standard predictions shows that the model can compensate prediction errors across time. Looking at the left part of the table shows that local predictions can be closer to the ground truth but give integrated predictions less accurate. This is explained by compensation. Table I shows that a result has a MAE on integrated predictions >15%. In this case, we observe a non smooth camera displacement during player motion. This particular behavior generates the gap between predictions and ground-truth.

	MAE			MAE on IP		
Video	NO PEN	PEN	NO PP	NO PEN	PEN	NO PP
1	0.01	0.01	0.22	1.03	0.10	2.21
2	0.27	0.23	0.42	4.17	5.67	12.33
3	0.72	0.86	1.09	10.26	13.70	22.05
4	0.62	0.58	0.95	14.02	9.85	23.9
5	0.32	0.33	0.46	15.27	11.41	14.77
6	0.22	0.25	0.52	5.69	12.36	11.6
7	0.45	0.55	0.70	4.71	4.72	5.91
8	0.58	0.36	0.61	4.84	4.99	6.47
9	0	0.11	0.58	0	3.58	24.12
10	0.72	0.54	0.82	12.05	6.26	13.10
11	0	0	0.15	0	0	2.70
12	0.66	0.65	0.97	15.74	12	17.55
13	0.98	0.86	1.07	24.58	18.57	23.33
14	0.27	0.11	0.25	2.92	4.18	2.41
15	0.35	0.41	0.57	8.40	7.01	9.97
			MEAN	9.29	**7.62**	12.23
			MAX	24.58	**18.57**	24.12
			NUMBER OF MAE >15	3	**1**	4

Table I: Mean absolute error computed on simple predictions (MAE) and on integrated ones (MAE on IP) with the two versions of the model for every video in the database. The table shows that PEN is the best model on this database and gives good tracking results on 14 videos (MAE on IP $\leq$ 15). In this table, the values are normalized to percentages of the total field. Videos where MAE on IP $\leq$ 15% of the total field are considered as well tracked.

IV. Conclusion and future works

In this paper, we have presented an optical flow based method for following the action in an amateur sports game. Our method deduces global motion needed to track action from segmented optical flow by applying a piecewise linear operator. The main advantage of this method is that accuracy does not depend on the camera position or lens quality as it is the case for methods tracking the ball by detection. Moreover, the computational cost is smaller than for the ball detection based methods and it mainly depends on computing optical flow complexity. Data can be automatically labeled with the global motion of the background.

Our method shows some limits in its capacity to track very long match sequences. With our automatic labelling procedure,

we are now able to construct a large database. A possible future work is to define a neural network architecture trained on such a database to predict motion. We can fit the network by using the model defined above to predict directions by analyzing optical flow vectors. Another way to improve the results is to combine detection based methods to this one to outperform the limits of each method.

ACKNOWLEDGEMENTS

This work was co-funded by Rematch Company, the Ministère en charge de l'Enseignement Supérieur, de la Recherche et de l'Innovation and ANRT who financed CIFRE theses.

Figure 3: Examples of good tracking results by the algorithm in videos n°2,7,14 and 15 in the database.

REFERENCES

[1] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.

[2] Tiziana D'Orazio, Cataldo Guaragnella, Marco Leo, and Arcangelo Distante. A new algorithm for ball recognition using circle hough transform and neural classifier. *Pattern recognition*, 37(3):393–408, 2004.

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[4] Elan Dubrofsky. Homography estimation. *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, 5, 2009.

[5] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1122–1132. IEEE, 2018.

[6] Paul W Holland and Roy E Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.

[7] Glenn Jocher, K Nishimura, T Mineeva, and R Vilariño. Yolov5. *Code repository https://github. com/ultralytics/yolov5*, 2020.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Jacek Komorowski, Grzegorz Kurzejamski, and Grzegorz Sarwas. Deepball: Deep neural-network ball detector. *arXiv preprint arXiv:1902.07304*, 2019.

[10] Haopeng Li and Markus Flierl. Sift-based multi-view cooperative tracking for soccer video. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1001–1004. IEEE, 2012.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[12] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[13] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of instance segmentation and object detection algorithms in pytorch. *Google Scholar*, 2018.

[14] Chris Poppe, Sarah De Bruyne, Steven Verstockt, and Rik Van de Walle. Multi-camera analysis of soccer sequences. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 26–31. IEEE, 2010.

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[17] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.

[18] Xinchao Wang, Vitaly Ablavsky, Horesh Ben Shitrit, and Pascal Fua. Take your eyes off the ball: Improving ball-tracking by focusing on team play. *Computer Vision and Image Understanding*, 119:102–115, 2014.

[19] Tharindu Romesh Fernando Warnakulasuriya, Xinyu Wei, Clinton Fookes, Sridha Sridharan, and Patrick Lucey. Discovering methods of scoring in soccer using tracking data. In *Proceedings of the 2015 KDD Workshop on Large-Scale Sports Analytics*, pages 1–4. KDD Workshop on Large-Scale Sports Analytics, 2015.

[20] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[21] Young Yoon, Heesu Hwang, Yongjun Choi, Minbeom Joo, Hyeyoon Oh, Insun Park, Keon-Hee Lee, and Jin-Ha Hwang. Analyzing basketball movements and pass relationships using realtime object tracking techniques based on deep learning. *IEEE Access*, 7:56564–56576, 2019.

[22] Xinguo Yu, Hon Wai Leong, Changsheng Xu, and Qi Tian. Trajectory-based ball detection and tracking in broadcast soccer video. *IEEE Transactions on multimedia*, 8(6):1164–1178, 2006.