Automatic Recognition of Sport Events from Spatio-temporal Data: An Application for Virtual Reality-based Training in Basketball

Alberto Cannavò, Davide Calandra, Gianpaolo Basilicò and Fabrizio Lamberti
Politecnico di Torino, Dipartimento di Automatica e Informatica, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Keywords: Machine Learning, Event Recognition, Virtual Reality, Sport Training.

Abstract: Data analysis in the field of sport is growing rapidly due to the availability of datasets containing spatio-temporal positional data of the players and other sport equipment collected during the game. This paper investigates the use of machine learning for the automatic recognition of small-scale sport events in a basketball-related dataset. The results of the method discussed in this paper have been exploited to extend the functionality of an existing Virtual Reality (VR)-based tool supporting training in basketball. The tool allows the coaches to draw game tactics on a touchscreen, which can be then visualized and studies in an immersive VR environment by multiple players. Events recognized by the proposed system can be used to let the tool manage also previous matches, which can be automatically recreated by activating different animations for the virtual players and the ball based on the particular game situation, thus increasing the realism of the simulation.

1 INTRODUCTION

In previous research, thanks also to recent advancements in tracking technology, the use of spatio-temporal data collected during matches or training sessions has grown significantly in many competitive sports (Richly et al., 2016). A number of solutions based on different sensing techniques have been presented in the literature, which allow to record the movement of the players and other equipment (a tennis ball, a baseball bat, etc.) at high sampling rates (von der Grün et al., 2011; Jiang and Yin, 2015; D'Orazio et al., 2010). The analysis of tracking data concerning the players, the ball, etc. can provide coaches with helpful insights about the game, which can be used for the automatic recognition of the opposing team's strategy (Varriale and Tafuri, 2016), the generation of commentaries for matches (Zheng and Kudenko, 2012), etc.

Based on these observations, this paper investigates the use of machine learning for the automatic recognition of players' activity – or actions – from spatio-temporal data for VR-based basketball applications (though information extracted could be exploited in other contexts, like those above). The paper builds on a previous work targeted to soccer (Richly et al., 2016). With respect to (Richly et al., 2016), in this paper new features are extracted, which permit a) to consider aspects that were not taken into consideration in that work, b) to integrate data not present in the reference dataset (like, for instance, the vertical position of the ball), and c) to account for different characteristics of basketball w.r.t. to soccer, with the final goal of improving recognition accuracy.

The recognition method proposed in this work has been integrated in an immersive VR tool to allow the visualization of animated reconstructions of previous basketball matches for tactic analysis and training. Specifically, events identified through machine learning are provided in input to the VR system, which uses them to activate proper player's animations.

2 BACKGROUND

A few methods have been experimented already for the automatic recognition of sport events. For instance, in (Zheng and Kudenko, 2012), inductive learning techniques are used for the automatic generation of commentaries for football matches within a management simulation game named Championship Manager. Three classification techniques (Decision Tree, KNN, and Naïve Bayes) are exploited to find the mapping between game states and commentaries. In (Teachabarikiti et al., 2010), an algorithm for tracking the players and the ball in tennis is proposed to enable automatic footage annotation. By analyzing the motion patterns of the players and the ball, the

Cannavò, A., Calandra, D., Basilicò, G. and Lamberti, F.
Automatic Recognition of Sport Events from Spatio-temporal Data: An Application for Virtual Reality-based Training in Basketball.
DOI: 10.5220/0007524203100316
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 310-316
ISBN: 978-989-758-354-4

algorithm is able to classify a player's action into either backhand and forehand stroke with high precision and recall rates. The authors of (McQueen et al., 2014) exploit players' tracking data to recognize offensive strategies in basketball through a linear SVM classifier and a rule-based algorithm. In (Richly et al., 2016), three machine learning approaches, namely SVM, KNN, and RF, are experimented for the purpose of classifying events in a soccer match, like passes or receptions. The dataset used therein refers to matches of the German Bundesliga, and contains the timestamp, the two-dimensional coordinates of the ball, a list of game events (e.g., fouls, substitutions, offsides, etc.) and player involved. Event classification is accomplished by working with several features computed by considering the raw position data for the ball. To train the classifiers, the dataset is annotated by manually identifying the events of interest in the footage of three matches.

By building upon the works found in the literature, this paper presents the design and evaluation of an improved technique for the automatic classification of sport events from spatio-temporal data. In particular, given the promising results reported in (Richly et al., 2016), this paper moves by considering the methodology developed in that work as a reference, and extends it to target a different sport, i.e., basketball. After having experimented the same algorithms and the same set of features used in the reference work on a dataset containing position data from National Basketball Association (NBA) matches, this paper additionally proposes a new set of features, which proved to significantly boost the performance of basketball event recognition and classification. Finally, the paper explores how automatically information extracted can be used to support the job of both coaches and players by enhancing the functionality of an existing VR-based tool for tactics analysis.

3 METHODOLOGY

This section describes the dataset as well as the features that have been developed/used in this paper.

3.1 Dataset

The original dataset refers to the 2015–16 season of the NBA (https://github.com/sealneaward/nba-movement-data/tree/master/data), and contains spatio-temporal data collected at 20 Hz. Data are structured in matches and actions (for a given match). For each action, the position of the ball and of the players is recorded. The dataset, stored as a *.csv* file, consists of the following values:

- $team_{id}$: identifier of the team to which player belongs to, -1 if the tracked object is the ball;
- $player_{id}$: identifier of the tracked object, -1 if the tracked object is the ball;
- x_{loc}, y_{loc}, z_{loc}: 3D spatial position of the tracked object (the z coordinate is provided only for the ball);
- $game_{clock}$: remaining time of the match;
- $shot_{clock}$: remaining time of the 24 seconds granted to a team to finalize an offensive action;
- $quarter$: quarter of the game;
- $game_{id}$: identifier of the match;
- $event_{id}$: identifier of the action in the game.

The coordinate system used for x_{loc} and y_{loc} is normalized in the $0-100$ and $0-50$ range, respectively for the x and y axis; the bottom-left corner is represented by point with $(0,0)$ coordinates. To create the annotated dataset, sports events were manually identified in the footage of the San Antonio Spurs vs Minnesota Timberwolves match that was played on December 23rd, 2015. Like in the reference work, passes and receptions were considered. Other events, like shots, dribbles, etc. were marked with the label "other". Part of the events belonging to the latter category were randomly deleted, in order to balance the frequency of the three events. At the end of the process, the annotated dataset included 180 entries per event category.

3.2 Features

According to the reference work, a sport event can be recognized in a dataset containing spatio-temporal data by analyzing the values of several features that characterize it. Features have been extracted by running a script on the above data. For each time t in the dataset, a vector is obtained containing a value for each feature. Features used in this work can be categorized in five groups. The first group contains the ("two-dimensional") features directly derived from the reference work. The remaining groups host the new features that have been introduced in this paper. In particular, the features in the second group are calculated by considering only the movement of the ball along the z axis; hence, they are referred to as "vertical". Features in the third group are those in the second group, but adapted to a "three-dimensional" space. For features in the fourth group, the position of the players is also considered; thus, they are called “players”’ features. Lastly, features in the fifth group are computed by aggregating data (and computing mean and variance values for them within given time windows): hence, they are referred to as “aggregated” features. The position of a tracked object o at time t will be represented as $p(o,t)$. Similarly, $p_x(o,t)$ and $p_y(o,t)$ will be used to refer to the position along the x and y axes. The distance between two consecutive positions will be defined as:

$$d(o,t_1) = p(o,t_2) - p(o,t_1) \quad (1)$$

where t_1 and t_2 are different time samples and $t_1 < t_2$.

3.3 Two-dimensional Features

The features in this group consider only the position of the ball in two dimensions (hence, a subscript 2D will be used). In the reference work, the z dimension was not considered because the soccer dataset contained only two coordinates for the ball.

3.3.1 Velocity

The two-dimension velocity, introduced since it is an indicator of the ball's momentum, is calculated by dividing the length of the direction vector $d(o,t_1)$ by the time interval between two adjacent samples:

$$Vel_{2D}(o,t_1) = \frac{|d(o,t_1)|}{t_2 - t_1} \quad (2)$$

3.3.2 Acceleration

The acceleration, like the velocity, was introduced as an indicator of the ball's momentum, and it is computed as:

$$Acc_{2D}(o,t_1) = \frac{Vel_{2D}(o,t_2) - Vel_{2D}(o,t_1)}{t_2 - t_1} \quad (3)$$

3.3.3 Acceleration Peaks

Given the sampling rate of the data, the same acceleration could be captured in consecutive time samples. Therefore, the authors of the reference work introduced two features referred to as acceleration peaks, that combine consecutive acceleration values by selecting the highest and the lowest ones among adjacent values, respectively. The computation of actual maximum and minimum peaks can be split in two steps. In the first step, the sum of two consecutive accelerations is computed ignoring negative and positive values by setting them to 0 for the computation of the former and the latter, respectively:

$$AP_{2D\_max}(o,t_2) = \sum_{x \in t_1,t_2} \max(0, Acc_{2D}(o,x)) \quad (4)$$

$$AP_{2D\_min}(o,t_2) = \sum_{x \in t_1,t_2} \min(0, Acc_{2D}(o,x)) \quad (5)$$

In the second step, in order to avoid the detection of a peak in two consecutive samples, the actual (real) acceleration peaks – $AP_{2D\_max_{real}}(o,t_2)$ and $AP_{2D\_min_{real}}(o,t_2)$ – are computed by setting them to $AP_{2D\_max}(o,t_2)$ and $AP_{2D\_min}(o,t_2)$ only if the value of the feature at time t_2 is higher than values at t_1 and t_3, otherwise they are set to 0.

3.3.4 Direction Change

This feature considers the variations in the trajectory of the ball during the game, taking into account the angle between two consecutive direction vectors. It was added to improve the recognition of event like passes or shoots characterized by an high value of this metric. The direction change $DC_{2D}(o,t_2)$ of object o at time t_2 is obtained by applying the arccos() function as follows:

$$DC_{2D}(o,t_2) = \arccos\left(\frac{d(o,t_1) * d(o,t_2)}{|d(o,t_1)| * |d(o,t_2)|}\right) \quad (6)$$

3.3.5 Distance to Target

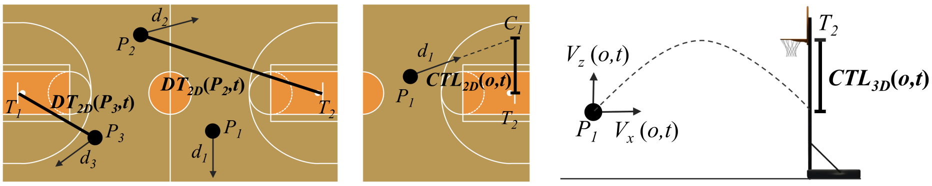
During a match, the ball should be thrown into one of the baskets (nets, in the reference work) in order to earn some points. Therefore, it is reasonable to assume that the ball moves towards one these targets. For this reason this metric is used to recognize passes from other shot. The distance of object o at time t from the target is calculated as:

$$DT_{2D}(o,t) = |p(o,t) - b(o,t)| \quad (7)$$

where *b(o,t)* represents the target position assigned depending on the direction of the ball w.r.t. to the x axis. This position could be either the point T_1 with coordinates (0, 25) if the ball moves towards the left side of the court, or T_2 with coordinates (100, 25) if the ball moves towards the right side, as shown in Figure 1a. The figure shows also different distances (represented by solid lines) computed depending on the direction of the ball (represented by an arrow at each data point). For object at point P_1, which is characterized by a horizontal velocity equal to 0, target cannot be determined; hence, the feature value is set to ∞.

3.3.6 Cross on Target Line

This feature is defined by considering the distance between the target and the position in which the ball would cross the end line should the current trajectory be maintained up to the line. Figure 1b shows a data point P_1 and its direction vector d_1. Should the



(a) (b) (c)

Figure 1: Calculation of a) Distance to Target, b) Cross on Target Line, and c) Cross on Target Line features.

ball continue to move without any direction change (dashed line), it would reach the end line in C_1. The distance between C_1 and the target position T_1 is the actual value of this feature. Position of C_1 can be calculated as:

$$\begin{pmatrix} b_x(o,t) \\ ctl \end{pmatrix} = p(o,t) + s * d(o,t) \tag{8}$$

where s is a factor that, if it is multiplied for the direction vector of the object o at time t and added to the position of the object o at time t, allows to reach the end line. From all of the above it is possible to compute $CTL_{2D}(o,t)$ as:

$$CTL_{2D}(o,t) = p_y(o,t) + d_y(o,t)\frac{b_x(o,t) - p_x(o,t)}{d_x(o,t)} \tag{9}$$

where the subscript identifies the axis considered.

3.4 Vertical Features

Features in this and in the following groups are those defined in the current work. In particular, this group contains some of the features in the previous group recomputed considering only the z coordinate: $Vel_V(o,t)$, $Acc_V(o,t)$, $AP_{V\_max_{real}}(o,t)$, and $AP_{V\_min_{real}}(o,t)$. The remaining features cannot be recalculated, since the single dimension considered does not allow to identify the direction of the ball.

3.5 Three-dimensional Features

In this group, features $Vel_{3D}(o,t)$, $Acc_{3D}(o,t)$, $AP_{3D\_max_{real}}(o,t)$, $AP_{3D\_min_{real}}(o,t)$, $CTV_{3D}(o,t)$ are calculated by considering the three coordinates. Hence, the subscript 3D is used. For $CTL_{3D}(o,t)$, a parabolic trajectory is assumed (as shown in Figure 1c), and the feature is computed as:

$$CTL_{3D}(o,t) = -\frac{1}{2}gt_{line}^2 + Vel_z(o,t)t_{line} + p_z(o,t) \tag{10}$$

where g is the gravity acceleration, $Vel_z(o,t)$ is the component along the z axis of $Vel_{3D}(o,t)$, and t_{line} is the time that is required for the ball to reach the end line; t_{line} is defined as:

$$t_{line} = \frac{b_x(o,t) - p_x(o,t)}{Vel_x(o,t)} \tag{11}$$

where $Vel_x(o,t)$ is the component of $Vel_{3D}(o,t)$ along the x axis.

3.6 Players' Features

This group contains two features that take into account the relationship between the position of the ball and the players. These features have been introduced because the way ball position changes in close proximity to a player could be a valid descriptor especially for some basketball events.

3.6.1 Ball-player Distance

This feature computes the distance between the ball and the closest player at time t. It is defined as:

$$BPD(o,t) = \left| p(o,t) - p_{player}(o,t) \right| \tag{12}$$

where $p(o,t)$ is the two-dimension position of the ball at time t and $p_{player}(o,t)$ is the two-dimension position of the closest player.

3.6.2 Team of Closer Player

This feature represents the team of the player closest to the ball at time t.

3.7 Aggregated Features

This group includes a set of features computed by aggregating consecutive samples. The aggregation considers the average and the variance values calculated in two time windows, named *before-window* and *after-window*. In this way, the aggregation allows to take into account the features' dynamics. The size of the two windows have been experimentally defined and includes 20 samples (i.e., one second) before and after the current time. The features considered for the aggregation are: $p_z(o,t)$, $Vel_V(o,t)$, $Acc_V(o,t)$, $DC_{2D}(o,t)$, $BPD(o,t)$.

Table 1: Recognition of basketball events using KNN.

	Pass	**Reception**	**Other**
Precision	0.69	0.68	0.93
Recall	0.65	0.67	1.00
F-measure	0.67	0.67	0.96
Accuracy	76.67%		

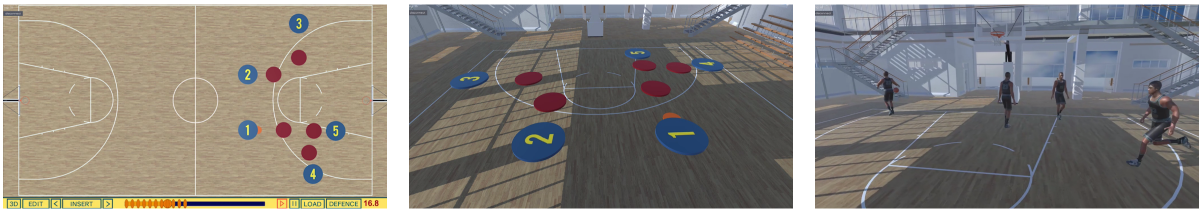
4 PERFORMANCE EVALUATION

Features described in the previous section have been used in combination with the three machine learning algorithms considered in (Richly et al., 2016). For every time *t*, a vector was created containing the values of all corresponding features. Each vector represents an event that occurs during the game and it is characterized by particular values of the defined features. For example, passes are characterized by a significant acceleration peak and presents a high value for the direction change feature, whereas in the case of receptions, the ball shows a strong negative acceleration and the distance with the closest player, probably the ball's owner, remains almost the same. The data science software platform named Rapidminer was used to run the algorithms. As said, the paper focused on the recognition of three events: pass, reception and other ball events though in basketball, rather than in soccer. In order to assess the quality of results achieved, accuracy, precision, recall and F-measure were calculated. To cope with the reduced size of the dataset, cross validation with 20 partitions and linear sampling were used. Evaluation was carried out by considering different combinations of the features in the five groups. Initially, only the first group was considered, to qualitatively compare results obtained on the new dataset with those in (Richly et al., 2016). Afterwards, the vertical and the players' features were added. The next experiment consisted in replacing the two-dimensional features with the three-dimensional ones. Lastly, the aggregated features were integrated. At every change in the set of features considered, the overall accuracy improved: from the initial value of 33.68% obtained when using only the first group of features (and comparable to that obtained in the reference work for soccer events), it reached a value of 76.67% when using the last set of features. Table 1 reports recognition results for each event obtained with KNN, which achieved the best performance.

5 APPLICATION SCENARIO

The method illustrated in the previous sections has been used to extend the functionalities of an existing tool for VR-based training in basketball. The tool, named *VR Playbook* (Cannavò et al., 2018), was designed to let coaches and players create tactics and visualize previous basketball games in an immersive environment. The VR Playbook tool offers coaches several graphics means for drawing a tactics in 2D with a tablet device by moving players and defining actions for them (passes, stops, throws, etc.) on a timeline (Figure 2a). The tool then creates the corresponding 3D animation that can be visualized at the same time by multiple players wearing VR headsets (Figures 2b and 2c). To this purpose, the timing and type of manually defined events are used to activate realistic players' animations which were previously recorded using motion capture. Tactics could also be saved (exported) and reloaded (imported) for later use. In the native implementation of the tool, in order to visualize the actions of a previous match coaches had to manually add players' events to the timeline, e.g., based on available game footage or by resorting to their memory. Players' trajectories could be either defined by drawing arrows on the touchscreen between the starting and ending points of a given action, or by adding many intermediate points to the timeline to avoid straight paths. Alternatively, they could load a dataset like the one used in this paper and recreate actual displacements. However, without annotations concerning events' timing and type, animations created would be poorly realistic, since positional data could only be used to activate a run cycle animation for players. In this paper, the devised methodology has been used to extract players' events from a dataset containing only spatio-temporal positional data and to store them in a format ready to be parsed and imported in the VR Playbook tool. In this way, the quality (realism) of the simulation can be improved, since the exact time a given animation shall begin/end is automatically defined, and a more correct relationship between the players' hands and the ball can be identified (and used for blending the run and pass animations). The integration of the devised methodology (the module named Event Recognizer) in the architecture of the VR playbook tool is depicted in Figure 3. It can be easily observed that integration is transparent to the users, since automatically extracted events are treated as manually defined ones, and coaches are allowed to further modify them using the tablet-based interface.

An example of the quality of animations that could be created using only dataset's raw data is given in Figure 4a. Improvements that could be obtained using the proposed automatic event recognition are illustrated in Figure 4b. A video is also available for download at https://goo.gl/ucDzH7.



(a) (b) (c)

Figure 2: VR Playbook tool: a) tablet interface for drawing tactics, and b)-c) animations displayed on VR headsets.

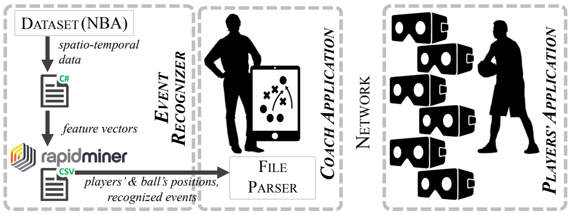
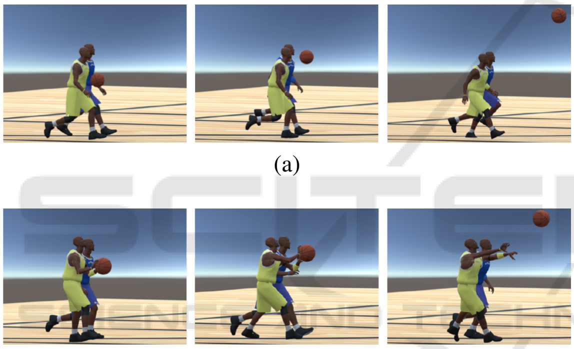


Figure 3: Integration of the devised event recognition methodology into the VR Playbook tool.



(a)

(b)

Figure 4: Frames of a 3D animation created using a) only raw positional data, and b) automatically recognized events.

6 CONCLUSIONS

Results reported in this paper confirm the suitability of machine learning techniques for the identification of small-scale sport events in spatio-temporal data collected during basketball games. In particular, features leading to good performances in the considered conditions are identified. Besides quantitative measurements concerning the accuracy of event recognition, preliminary evidences on the effectiveness of the devised methodology have been also collected through qualitative observations on the realism of animations that can be generated by integrating automatic event recognition in a computer animation tool. Future work will be devoted to the exploration of new features and classification methods (e.g., based on deep learning) as well as the recognition of other small-scale basketball events (like throws, screens, cuts, etc.) and of large-scale phenomena occurring during the game (e.g., to predict dangerous actions, to identify tactics, to spot mistakes made by a player in executing a tactic, etc.). The introduction of these new aspects and the development of improved techniques for animation blending could help to further enhance the quality of the animations that can be produced, making (VR-based) visualization systems suitable also for sport applications different than training. Moreover, a user study will be planned with coaches and players of a basketball team to validate the effectiveness of the VR training system.

ACKNOWLEDGEMENTS

This work has been partially supported by VR@Polito initiative. The authors wish to thank Francesco Raho, the technical manager of the Auxilium CUS basketball Torino's youth sector, Italy.

REFERENCES

Cannavò, A., Musto, M., Pratticò, F. G., Raho, F., and Lamberti, F. (2018). A participative system for tactics analysis in sport training based on immersive virtual reality. In *4th Workshop on Everyday Virtual Reality*.

D'Orazio, T., Leo, M., Mazzeo, P. L., and Spagnolo, P. (2010). Soccer player activity recognition by a multivariate features integration. In *7th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*.

Jiang, W. and Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *23rd ACM Int. Conf. on Multimedia*, pages 1307–1310.

McQueen, A., Wiens, J., and Guttag, J. (2014). Automatically recognizing on-ball screens. In *2014 MIT Sloan Sports Analytics Conference*.

Richly, K., Rohloff, T., Bothe, M., and Schwarz, C. (2016). Recognizing compound events in spatio-temporal football data. In *Int. Conf. on Internet of Things and Big Data*.

Teachabarikiti, K., Chalidabhongse, T. H., and Thammano, A. (2010). Players tracking and ball detection for an automatic tennis video annotation. In *11th Int. Conf. on Control Automation Robotics & Vision*.

Varriale, L. and Tafuri, D. (2016). Technology for soccer sport: The human side in the technical part. In *Int. Conf. on Exploring Services Science*.

von der Grün, T., Franke, N., Wolf, D., Witt, N., and Eidloth, A. (2011). A real-time tracking system for football match and training analysis. In *Microelectronic Systems*, pages 199–212. Springer.

Zheng, M. and Kudenko, D. (2012). Automated event recognition for football commentary generation. In *Interdisciplinary Advancements in Gaming, Simulations and Virtual Environments: Emerging Trends*, pages 300–315. IGI Global.