

Article

Recognizing Events in Spatiotemporal Soccer Data

Victor Khaustov *  and Maxim Mozgovoy * 

School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan

* Correspondence: d8201106@u-aizu.ac.jp (V.K.); mozgovoy@u-aizu.ac.jp (M.M.)

Received: 13 October 2020; Accepted: 10 November 2020; Published: 13 November 2020



Abstract: Spatiotemporal datasets based on player tracking are widely used in sports analytics research. Common research tasks often require the analysis of game events, such as passes, fouls, tackles, and shots on goal. However, spatiotemporal datasets usually do not include event information, which means it has to be reconstructed automatically. We propose a rule-based algorithm for identifying several basic types of events in soccer, including ball possession, successful and unsuccessful passes, and shots on goal. Our aim is to provide a simple procedure that can be used for practical soccer data analysis tasks, and also serve as a baseline model for algorithms based on more advanced approaches. The resulting algorithm is fast, easy to implement, achieves high accuracy on the datasets available to us, and can be used in similar scenarios without modification.

Keywords: event detection; spatiotemporal analysis; soccer analytics

1. Introduction

Sports analytics can serve as a rich source of data for a variety of machine-learning tasks. Widely available sports-related datasets range from collections of individual players' performance indicators and historical game stats, to detailed event logs of particular matches. Recent advancements in object tracking technology have enabled sports analysts to collect spatiotemporal information, reflecting athletes' movements over time. Resulting spatiotemporal datasets have been used in numerous research tasks, including similar play sequences retrieval [1] and identification of defensive strategies [2] in basketball, and shot prediction in tennis [3]. In soccer, spatiotemporal datasets have been used to analyze team formations and play styles [4], as well as learning coordinated defensive strategies [5].

Most available spatiotemporal sports datasets, to the best of our knowledge, are organized as sequences of game field snapshots that include athletes' locations at specific points of time. However, for many types of research projects, information about game events is also crucial. For example, in the case of team sports games (such as soccer, basketball, or rugby) it might be important to know, for example, which athlete is currently possessing the ball, whether a snapshot with a ball in the air belongs to a pass or a shot on goal sequence, and so on. One realistic scenario where such information is necessary is the development of a case-based reasoning AI system for playing a sports game [6]. The AI needs to learn actions performed by the athletes in specific game situations, so both these actions and situations must be reconstructed from a series of snapshots, contained in a dataset. Other possible applications of this kind of data are sports analytics and game visualization [7].

In this work, we discuss the procedures we employ for the automatic detection of ball events in spatiotemporal soccer datasets. By using them, it is possible to extend existing datasets with information about player movements, ball possession, passes, shots on goals, and tackles, necessary for a variety of sports analysis-related tasks. Our primary goal is to ensure the simplicity and accuracy of the methods, and their easy integration into a dataset processing pipeline. Therefore, in most

cases, we rely on experts' knowledge of the game to define the explicit rules for detecting events instead of trying to extract those rules from the data. In addition, this approach helps to avoid overfitting, which might occur with machine learning-based models trained on insufficiently large datasets, common in sports analytics.

This paper is organized as follows. The "Background and Related Works" section briefly discusses the traits of soccer spatiotemporal datasets available today. We note the widespread absence of event markup and examine existing approaches to generate it. We discuss the newest works in this area, which are mostly focused on application and evaluation of specific methods for event recognition. The "Experimental Setup" section provides a description of the datasets we use and their internal structure. We also define the specific list of events we are interested in, and note their properties. The next section titled "Game Event Detection Pipeline" discusses specific algorithms and rules used to detect each event type. Sufficient details are provided to recreate independent implementation, if necessary. The "Event Detection Quality Evaluation" section is dedicated to the analysis of the accuracy of the proposed approach. We report results independently for each of our datasets and estimate test accuracy using the F_1 score. The final sections, the Discussion and Conclusion, restate our contribution. We show how the obtained results are connected with our initial goals, what the advantages and limitations of our approach are, and outline possible further research directions.

2. Background and Related Works

Sports provide a vast amount of diverse information, ranging from statistical tables reflecting the performance of individuals and teams to video streams of competitions. However, different sources of information provide different types of data, and it is not always possible to combine them to obtain a comprehensive picture of certain phenomena.

The reasons for such fragmentation are both technological and legal. For example, the English Premier League compiles and publishes basic statistical facts about their soccer players, and, in general, such data is often reported in newspapers and various "yearbooks". Some research works are focused on analyzing soccer video streams and detecting the events directly from them [8,9]. However, video recordings of complete matches of a particular event might be hard to obtain, and their distribution is typically strictly controlled by the copyright holder. Soccer video recordings that capture the whole field are rarely available at all. Often the content of a particular dataset reflects the collectors' understanding of what constitutes "interesting" information. For example, an often-cited F24 soccer feed [10] provided by Opta Sports [11] includes specific types of in-game events, manually annotated by the experts [12]. Copyright issues limit the availability of such datasets, and the question of ownership in borderline cases remains open [13].

We are specifically interested in spatiotemporal soccer player tracking data with ball-event markup. Player tracking data is available from several providers, such as [StatsPerform.com](https://www.statsperform.com), [DataStadium.co.jp](https://www.datastadium.co.jp), and [Chyronhego.com](https://www.chyronhego.com). Typically, it is obtained by the software-assisted digitization of video streams, recorded by several fixed cameras, installed at a stadium [14]. This technological process presumes no additional post-processing, therefore producing player and ball coordinates only. On the other hand, ball-event datasets, such as the aforementioned Opta's F24 feed or the *Soccer match event dataset* [15] are typically focused on events, and do not provide complete player tracking data. In cases where manual event annotation is available, time synchronization might be inaccurate, and errors in player attribution are common [16].

Event detection in soccer datasets is the subject of several recent research works. A survey made by Gudmundsson and Horton [17] lists numerous tasks related to the spatiotemporal analysis of team sports. However, as of early 2016, only a few attempts have been made to deal with ball-event processing. Furthermore, these works are dedicated either to categorizing and labeling known events [18,19] or predicting future events [20,21]. Certain approaches to automatic event detection have been proposed only in more recent papers [16,22,23].

Richly et al. [16,22] applied several different machine learning-based methods to recognize four event types (pass, reception, clearance, and shot on goal) in a spatiotemporal soccer dataset. They prepared a “gold standard” manual markup comprising 194 events within 8:47 min of active game time and used it for training and testing different methods, including Support Vector Machine, K-Nearest Neighbors, Random Forest, and Artificial Neural Networks. Their best results were achieved with neural networks, yielding a precision of 89% and a recall of 90%. Interestingly, a significant quality improvement was obtained by applying a smoothing filter that reduced the original 25 Hz sampling rate of game recordings to 10 Hz. It is also worth mentioning that the resulting event markup did not include extended event information, such as a passer’s and receiver’s identity in the case of a pass event. Finally, it can be argued that the training set was relatively small (it included only seven shots on target, for example), so a more thorough evaluation might be necessary to assess the quality of the proposed approach in practical scenarios.

Morra et al. [23] experimented with a much larger Soccer Event Recognition (SoccerER) dataset comprising 500 min of gameplay. This dataset consists of artificial (simulated) soccer matches obtained with an open-source Gameplay Football engine [24]. This approach made it possible to evaluate their algorithm on thousands of events, annotated with perfect accuracy. However, such data should be treated as an approximation of a real scenario, where camera jitters and imperfect player tracking methods often lead to artifacts that can be observed in digitized recordings of actual soccer matches. The method for detecting game events was based on a set of handcrafted rules, expressed with temporal logic statements. These rules were implemented using the ETALIS library for Prolog. The work of Morra et al. dealt with a more complex set of events, so a direct comparison of the obtained results with the ones obtained by Richly et al. [16,22] would be inaccurate. Still, the authors reported an improvement of precision at 96%, and to recall at 93%.

3. Experimental Setup

As noted by Morra et al. [23], it is difficult to compare results obtained in different research works due to differences in experimental settings, types of events, and used datasets. Therefore, details of the experimental setup are essential, as they might have a significant impact on results. In addition, details of our datasets show what can be expected from acquired player tracking data in typical cases.

We work with two distinct spatiotemporal soccer datasets, obtained by tracking players with several fixed cameras, and subsequent digitization of recorded video streams (see Table 1). The first dataset (“DS”) represents five full matches of the Japanese J1 league captured in 2011. These recordings accurately capture the course of the game, and even significant pauses within the game are not removed. Player tracking is accurate in general, but occasional jitters do occur, so one may observe sudden jumps of player and ball objects. Such situations usually happen when many players are close to each other, particularly during free and corner kicks. Each captured frame is annotated with two additional binary attributes: *Ball owning team* (home/away) and *ball status* (dead/alive), so we know which team is the attacker, and whether the game is stopped by the referee.

Table 1. Experimental datasets.

Name (Provider)	Sequences	Total Time	Frames per sec	Additional Features
DS (Data Stadium)	5	320 min (5 games)	25	<i>ball owning team</i> and <i>ball status</i> fields tracking data of a referee z-coordinate of the ball timewasting periods are included player names and roles are known
ST (Stats Perform)	7578	2220 min (45 games)	10	attacking team is always on the left no “dead” segments (stopped by referee) teams are always complete (11 players) player data is anonymized

The second dataset (“ST”) [5] consists of a large number of short game episodes (from 5 to 150 s), taken from recent matches played in a top European league. An episode starts when a certain team gets possession of the ball, and ends when the team loses control of the ball. Player movements are smooth, and jitters are rare. However, this dataset has few shots on goal and few unsuccessful passes. Occasionally the outcome of the attacking team’s last action can be analyzed, but episodes ending with the ball still in the air are also common.

The ST dataset is organized as a collection of independent sequences, representing game episodes. Each sequence contains a list of frame objects, organized as follows:

$$(G_x, G_y)(P1_x, P1_y) \dots (P10_x, P10_y)(G'_x, G'_y)(P1'_x, P1'_y) \dots (P10'_x, P10'_y)(B_x, B_y).$$

A frame starts with the (x, y) coordinates of the goalkeeper of the team currently possessing the ball. After the goalkeeper coordinates, the coordinates of the field players of the same team are listed. There is no predefined order of players in this list, but the same order is preserved throughout the episode, so it is possible to trace the trajectory of a certain player in the given episode. The total number of field players is always 10, which means that only episodes featuring complete teams are included in the dataset. The next block describes the opposing team in the same manner. Finally, there are (x, y) coordinates of the ball. All coordinate values lie within a range of $[-52.5 \text{ m}, +52.5 \text{ m}]$ along the x -axis, and $[-34.0 \text{ m}, +34.0 \text{ m}]$ along the y -axis.

We convert the DS dataset into the same representation, though the original format contains more information (see Table 1).

Current literature shows that there is no universal agreement on the list of events that need to be identified in such player-tracking data, so different authors develop their own schemes suitable for their goals. For our ultimate goal (the development of a case-based reasoning soccer AI system), the following events were identified:

- *Successful pass event.* A player successfully passed the ball to an identified teammate;
- *Unsuccessful pass event.* A player tried to make a pass to an identified receiver, then the ball left the field or was intercepted by the other team. Note that “clearance” events (when the players kick the ball away from their own goal line) often fall into this category;
- *Shot on goal.* A player attempted a shot on goal, characterized by a certain target point.

In addition, we identify (1) the player currently possessing the ball, and (2) player movements, defined as the speed and direction of the given player calculated with the required precision.

4. Game Event Detection Pipeline

4.1. Player Movements Analysis

In our work, we are only interested in the basic approximation of actual player movements. We divide an input game segment into equal intervals of user-specified duration and calculate player velocity components v_x and v_y using Equations (1) and (2) according to their positions $x(t)$, $y(t)$ at the beginning and end of each time interval $[t_0, t_1]$. Note that this approach represents player movements with straight lines, approximating the general trajectory (see Figure 1):

$$v_y = \frac{y(t_1) - y(t_0)}{t_1 - t_0} \quad (1)$$

$$v_x = \frac{x(t_1) - x(t_0)}{t_1 - t_0}. \quad (2)$$

Even such a simple method allows us to study player movements with arbitrary precision and analyze their basic traits. In particular, we were able to distinguish real teams from teams comprised

of rule-based AI bots by analyzing probability distributions of player movement directions in different zones of the game field [25].

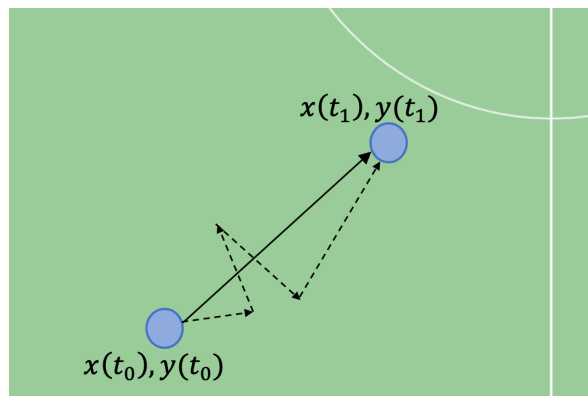


Figure 1. Approximation of player movement speed and direction. Solid lines represent constructed approximation; dashed lines correspond to the original player trajectory.

However, it should be mentioned that in general, the analysis of player movements is a more complicated process. In raw data captured by player tracking systems, jitter is inevitable, so certain smoothing algorithms are required. The choice of these algorithms, in turn, is not a trivial process, since sudden changes in speed and direction are very common in soccer, so smoothing may cause undesirable distortions [26]. One way of removing them is to use the Gaussian smoothing kernel [27], where x and y components of the trajectory are processed separately and treated as one-dimensional time-dependent signals, according to Equations (3) and (4):

$$x'(t) = \frac{1}{2N_F + 1} \sum_{i=-N_F}^{N_F} x(t+i) \cdot G(i) \tag{3}$$

$$y'(t) = \frac{1}{2N_F + 1} \sum_{i=-N_F}^{N_F} y(t+i) \cdot G(i). \tag{4}$$

Here $2N_F + 1$ denotes the width of the kernel, x' and y' are the smoothed components of the trajectory, and x and y are the components of the raw trajectory. G is the set of Gaussian coefficients defining the shape of the kernel.

4.2. Ball Possession

Both our datasets contain information about the team currently possessing the ball. Each frame in the DS dataset has a special attribute indicating a ball-owning team. Each recording in the ST dataset corresponds to a short game episode, where the team on the left-hand side possesses the ball. However, it is also necessary for us to know which player of the ball-possessing team dribbles the ball at a given moment. In general, three separate cases have to be identified:

1. The player dribbles the ball, so the ball is located in the immediate vicinity of the player;
2. The ball is outside the immediate reach of any player, but it can be treated as “being possessed” by a certain player. For example, a player might kick the ball out of immediate reach, but still keep it under control;
3. The player performs a pass or attempts a shot on goal. During this event, the ball is not possessed by any player, but we can still treat the current team as possessing the ball, until it is intercepted by the other team.

The first case is easy to identify since it is a matter of simply checking the distance between the ball and the closest player of the ball-owning team. If this distance is shorter than a certain threshold

(*VicinityThreshold*, see Table 2), the player is treated as possessing the ball. However, if several teammates are located near the ball, we give the preference to the player who controlled the ball on the previous frame to avoid unwanted changes in ball possession.

Table 2. Initial experimental setup: Parameters.

Parameter	Value	Description
<i>VicinityThreshold</i>	1.0 m	Minimal distance between the ball and the ball-possessing player
<i>GracePeriod</i>	1.0 s	Time required for a player to be considered a ball receiver
<i>GoalpostDistance</i>	5.0 m	Minimal distance from a goalpost for a kick to be considered a pass
<i>MinFailedPassLength</i>	2.5 m	Minimal ball travel distance of an unsuccessful pass
<i>MinTrChangeAngle</i>	12.5°	Minimal change in trajectory direction considered “significant”
<i>MinSpeedChangeFactor</i>	1.5	Minimal change in ball speed considered “significant”

The second case is treated by our pipeline as possession with an additional “*ball far away*” flag set. If the ball leaves the immediate vicinity of a ball-possessing player p and eventually is possessed by the same player again, we mark the whole segment as the player’s possession. Special handling of such segments allows us to perform a finer-grained analysis of possible actions in specific game moments. Players possessing a faraway ball cannot perform passes and shots on goal until they reach the ball again.

The third case (passes and shots on goal) requires more complex detection procedures, which will be covered in the subsequent section.

4.3. Detecting Passes and Shots on Goal

When a ball-possessing player attempts to pass the ball to a teammate, we register the occurrence of a pass event. Some passes are successful, while others end with the ball out of bounds or intercepted by the opposing team. A passer, a target receiver, and a pass result (successful/failed) comprise pass event markup in our system.

A simple approach for detecting a pass, therefore, can be based on detecting two consecutive basic events: (1) The ball leaves the immediate vicinity of a ball-possessing player; (2) the ball goes out of bounds or comes into the possession of another player (either a teammate or an opponent). In the case of the ST dataset, detecting whether the ball is intercepted by a certain player is not an entirely straightforward process due to the absence of a z-coordinate of the ball. Data does not show whether the ball approaches the player at a low trajectory or flies high above the player’s head. A sharp change of the ball trajectory or speed near a player is a good indication of pass reception. However, passes that do not significantly alter ball movement are also common (for example, defenders often make forward passes to midfielders moving in the same direction). Thus, we use both change of ball speed/trajectory and the presence of a certain “grace period” (see Table 2) when the ball is near the potential receiver as indications of a pass event (see Listing 1).

A ball going out of bounds is another indication of an unsuccessful pass event. Such passes have to be distinguished from shots on goal. We believe that for most practical tasks, it is enough to treat an event, which ends with a ball passing closer than a certain distance *GoalpostDistance* (see Table 2) from the nearest goalpost, as a shot on goal, and all other “out of bounds” situations as resulting from unsuccessful passes. Since shots are characterized by a shot target point in our system (a single value, representing an offset from the goal center), we note the specific location where the ball crosses the goal line. If this point is outside the goal, we correct the value by moving it to the nearest goalpost and treat this new target as the true intention of the attacker.

Finally, we need to identify a target receiver of an unsuccessful pass. This can be a challenging task even for a human observing the game, especially using 2D visualization. In the current system, we use the following heuristics: An intended receiver is the teammate closest to the ball at the moment when it has left the field or has been intercepted by the opponent. In addition, we filter out all

opponent-intercepted passes shorter than *MinFailedPassLength* (see Table 2). Such situations are treated as tackles, and the ball is merely transferred from one team to another without a pass or shot attempt.

As a result, in the current version of the system, we implement the following procedure for detecting passes and shots on goal (see Listing 2).

Listing 1. Detecting changes in ball possession.

Require: ball enters vicinity of a non ball-possessing player p'

```

1: function ISPOSSESSIONCHANGED(currFrame)
2:   currFrame ← CURRENTFRAME()
3:   prevFrame ← GETFRAMEAT(currFrame − 1)
4:   nextFrame ← GETFRAMEAT(currFrame + 1)

5:   prevSpeed ← CALCULATEBALLSPEED(prevFrame, currFrame)
6:   nextSpeed ← CALCULATEBALLSPEED(currFrame, nextFrame)
7:   speedChange ← MAX(nextSpeed, prevSpeed) / MIN(nextSpeed, prevSpeed)
8:   if speedChange > MinSpeedChangeFactor then
9:     return true                                     ▷ speed changed

10:  prevDir ← CALCULATEBALLDIRECTION(prevFrame, currFrame)
11:  nextDir ← CALCULATEBALLDIRECTION(currFrame, nextFrame)
12:  if |nextDir − prevDir| > MinTrChangeAngle then
13:    return true                                     ▷ trajectory changed

14:  if ball is still possessed by  $p'$  at currFrame + GracePeriod then
15:    return true                                     ▷ ball possessed by  $p'$  after grace period

16:  return false                                     ▷ no change in ball possession so far

```

Listing 2. Detecting passes and shots on goal.

```

1: function DETECTPASSORSHOT()
2:   if |Bally| > 34.0 then                             ▷ ball crossed the touchline
3:     return VERIFYFAILEDPASS()

4:   if |Ballx| > 52.5 then                             ▷ ball reached the goal line
5:     if DISTANCE(Bally, 0) < GoalpostDistance + GoalLength / 2 then
6:       return shot on goal
7:     else
8:       return VERIFYFAILEDPASS()

9:   if ball is within vicinity of another player  $p'$  then
10:    if ISPOSSESSIONCHANGED() is false then           ▷ see Listing 1
11:      return no event detected

12:    if  $p'$  is a teammate then
13:      return successful pass
14:    else
15:      return VERIFYSHOT()

16: function VERIFYFAILEDPASS()
17:   if pass distance is longer than MinFailedPassLength then
18:     return unsuccessful pass                             ▷ See Figure 2a
19:   else
20:     return no event detected

21: function VERIFYSHOT()
22:   if  $p'$  is in the goal area and ball trajectory line crosses the goal line
23:     not further than GoalpostDistance from the goalpost then
24:       return shot on goal                             ▷ See Figure 2b
25:   else
26:     return VERIFYFAILEDPASS()

```

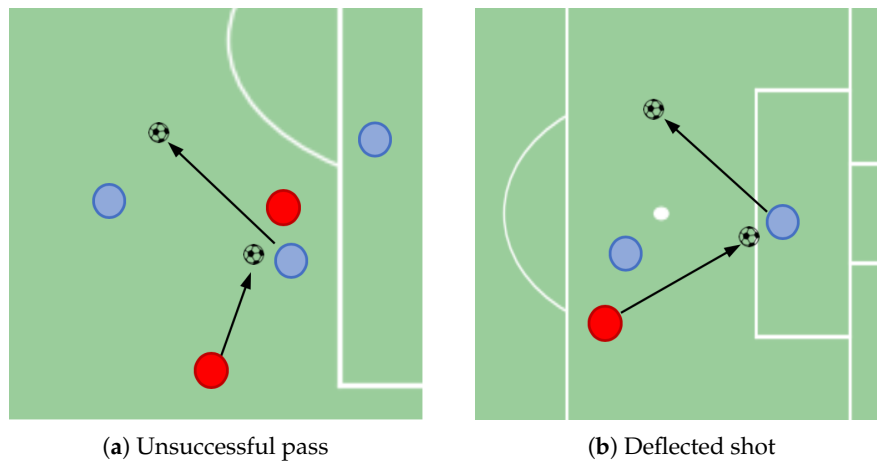


Figure 2. A scheme of an unsuccessful pass and a deflected shot.

5. Event Detection Quality Evaluation

The most complicated part of event markup in our pipeline is the detection of passes and shots. Thus, we perform a brief evaluation to estimate its accuracy.

We created a “gold standard” markup by watching the games that comprise our datasets in a 2D soccer simulator and manually annotating successful passes, unsuccessful passes, and shots on goal. In the case of the ST dataset, we analyzed the 40 longest episodes, corresponding to 60 min of playing time in total (see Table 3). In the case of the DS dataset, all five matches were annotated. As manual annotations are difficult to synchronize with events, we consider events as correctly recognized if they fall within a $[-0.5\text{ s}, +0.5\text{ s}]$ interval of the corresponding “gold standard” event.

Table 3. Annotated events.

Event	Quantity	
	ST	DS
Successful pass	1097	2910
Unsuccessful pass	95	331
Shot on goal	8	97

Since the proposed algorithm relies on specific parameter values, listed in Table 2, fine-tuning them is necessary to achieve high event detection performance. The initial experimental setup is based on manually chosen values, reflecting our general understanding of soccer and observations of the gold standard markup process.

To reveal the optimal combination of parameter values, we apply a greedy search routine. It evaluates the algorithm on the set of parameters $\{p_1, \dots, p_i, \dots, p_n\}$, where the value of p_i is iterated over a predefined range, while the rest of the values remain constant. The same process is repeated for each parameter in the set. This approach is feasible in our case because most parameters are loosely related and affect different types of situations in the game. For example, both *MinTrChangeAngle* and *MinSpeedChangeFactor* values affect the ability of the system to recognize changes in ball possession. However, they are introduced to deal with different types of game episodes (see Listing 1), so an optimal value of one parameter should increase the overall performance of the whole event detection algorithm. It is also easy to choose reasonable ranges and loop steps due to physical constraints of soccer.

The quality of event detection can be estimated with recall, precision, and F_1 score values, calculated according to Equations (5)–(7). The search routine iterates over a range of permissible values of an individual parameter, while the rest are fixed at their initial values, specified in Table 2.

We perform this routine for each parameter except *GoalpostDistance*, which affects shot on goal events only. Since the initial setup already provides optimal recognition of shots on goal, we keep the value of *GoalpostDistance* unchanged. It should also be mentioned that we perform a search for optimal parameter values using the ST dataset only. The DS dataset contains significant player jitter, and thus cannot serve as a reliable ground for fine-tuning. In contrast, the ST dataset consists of short game fragments played by different teams in different matches, serving as a reliable sample of episodes found in a typical soccer game. In our experiments, 70% of the annotated ST dataset is used to determine the optimal parameter values, while the remaining 30% of data is reserved for the evaluation:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{5}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{6}$$

$$F_1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

The evaluation routine (the source code for event recognition and parameter tuning is available at github.com/vi3itor/soccer-event-recognition) allows us to make the following observations (see Figure 3). Any choice of *MinFailedPassLength* and *MinSpeedChangeFactor* within the test ranges have virtually no effect on the resulting performance. The optimal value of *MinTrChangeAngle* is approximately 0.1, all higher values cause errors in pass events recognition. The optimal values of *GracePeriod* and *VicinityThreshold* are achieved inside their respective ranges and any deviation from the optimum increases recognition errors. On the other hand, nearly optimal results (with F_1 score higher than 0.9) are obtained on a wide range of values. The optimal choice of parameters is provided in Table 4.

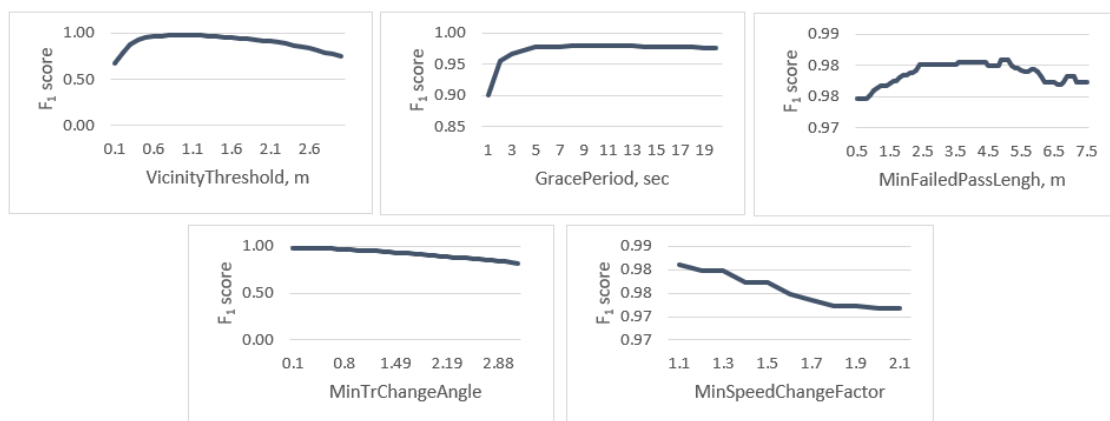


Figure 3. The influence of parameter values on event recognition quality.

Table 4. Optimal parameter values.

Parameter	Value
VicinityThreshold	0.9–1.2 m
GracePeriod	0.5–2 s
MinFailedPassLength	0.9–7.5 m
MinTrChangeAngle	5–25°
MinSpeedChangeFactor	1.1–1.5

The resulting evaluation of our algorithm, performed with the optimal set of parameters on both ST and DS datasets, is shown in Table 5. Since the performance of the algorithm is stable on a wide range of input parameter values, we believe it is able to achieve comparable results on other spatiotemporal soccer datasets.

Table 5. Accuracy of event detection.

Event	ST			DS		
	Precision	Recall	F ₁ Score	Precision	Recall	F ₁ Score
Successful pass	0.9991	0.9973	0.9982	0.9218	0.9470	0.9342
Unsuccessful pass	0.8318	0.9368	0.8812	0.8182	0.9013	0.8577
Shot on goal	1.0	1.0	1.0	0.9336	0.9607	0.9469

We should note that the DS dataset is more difficult to analyze both manually and automatically. This happens because of a lower tracking accuracy, especially in the case of overlapping players. DS tracking data is obtained with TRACAB technology available in 2011, the current Generation 5 TRACAB system provides more accurate results in such scenarios [28].

Generally, errors in event recognition occur in borderline cases. For example, there are situations where a trajectory of a flying ball is slightly changed after passing a teammate's vicinity. The algorithm considers this change insignificant, while a human expert recognizes it as a one-touch pass made by a teammate. Similarly, there are discrepancies in recognizing intended ball receivers in cases of unsuccessful long-distance passes.

As seen in Table 3, the ST dataset contains few unsuccessful passes and shots on goal. Thus, the evaluation of detection accuracy for these events is not as reliable as for the DS dataset. The lower accuracy of event detection for the DS dataset can be explained by considerable player and ball jitter. The DS dataset is recorded at a 25 Hz rate, and there is no trajectory smoothing, so our algorithms occasionally cannot project actual player movements.

6. Discussion

Event recognition in player tracking data is the subject of several research works. The best results to date have been reported in Richly et al. [22] and Morra et al. [23]. However, it seems that obtaining the most accurate results was not the main goal of these works. Both papers were focused on the evaluation of specific methods as seen from the conclusions made by their respective authors: *"the results showed that neural networks present a viable model to detect events in soccer data"* [22]; *"we have shown that ITLs (interval temporal logics) are capable of accurately detecting most events from positional data"* [23]. Therefore, high accuracy demonstrates the versatility of the suggested methods and provides a firm ground for their use in similar tasks.

Our primary goal is to develop a procedure that would provide a quick and accurate event markup of specific soccer datasets we use. It is hard to say how well the same approach would work in other team sports games. However, in the case of soccer, the obtained results are very close to the "gold standard" manual markup, so we consider our algorithms ready for practical use. We have to repeat that specific precision/recall values reported in the present work should be treated as dependent on the experimental setup, and specific types of events in particular.

In general, we should note that our rule-based approach possesses a number of advantages. It is simple, straightforward, and can be easily implemented in any conventional programming language required in a given project. It does not need a large annotated dataset for learning, which might be important for rare events such as shots on goal, where it is hard to collect a sufficient number of observations. It can be easily updated or modified, and it can serve as a baseline procedure for evaluating other algorithms based on more advanced methods.

One obvious problem with our approach is related to its dependency on specific parameter values listed in Table 2. Thus, the quality of event recognition might vary across datasets (corresponding to

teams of different skill levels, for example). However, present parameter values are chosen based on a general knowledge of soccer and demonstrate robust performance on wide value ranges, so we believe they are applicable to diverse collections of soccer matches.

Summing up, the resulting procedure is simple, fast, and accurate, and is able to recognize soccer events with comparable or higher precision than competing approaches. However, its flexibility is limited: while its operation can be adjusted with fine-tunable parameters, any changes in the list of supported events might require a significant code update.

7. Conclusions

We designed and implemented a rule-based algorithm for event detection in a spatiotemporal soccer dataset. Our method achieved high accuracy on two datasets and most probably could be used in other similar scenarios without modification. The proposed scheme could also supply baseline event recognition quality indicators for subsequent research projects. Similar techniques could be used to recognize other significant game events, including offsides, free kicks, and corner kicks. Further research directions may include support for other event types and adaptation of our algorithm for other team sports, such as basketball or ice hockey.

Author Contributions: Conceptualization, V.K. and M.M.; data curation, V.K.; writing—original draft preparation, V.K.; writing—review and editing, V.K. and M.M.; visualization, V.K.; supervision, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We thank Jeremy Perkins and Allan Nicholas for English language editing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sha, L.; Lucey, P.; Zheng, S.; Kim, T.; Yue, Y.; Sridharan, S. Fine-Grained Retrieval of Sports Plays using Tree-Based Alignment of Trajectories. *arXiv* **2017**, arXiv:1710.02255.
2. Tian, C.; De Silva, V.; Caine, M.; Swanson, S. Use of Machine Learning to Automate the Identification of Basketball Strategies Using Whole Team Player Tracking Data. *Appl. Sci.* **2019**, *10*, 24. [[CrossRef](#)]
3. Wei, X.; Lucey, P.; Morgan, S.; Sridharan, S. Predicting shot locations in tennis using spatiotemporal data. In Proceedings of the 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Hobart, Australia, 26–28 November 2013; pp. 1–8.
4. Bialkowski, A.; Lucey, P.; Carr, P.; Yue, Y.; Sridharan, S.; Matthews, I. Identifying Team Style in Soccer Using Formations Learned from Spatiotemporal Tracking Data. In Proceedings of the 2014 IEEE International Conference on Data Mining Workshop, Shenzhen, China, 14 December 2014; pp. 9–14.
5. Le, H.M.; Yue, Y.; Carr, P.; Lucey, P. Coordinated multi-agent imitation learning. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 7–11 August 2017; Volume 70, pp. 1995–2003.
6. Khaustov, V.; Mola Bogdan, G.; Mozgovoy, M. Pass in Human Style: Learning Soccer Game Patterns from Spatiotemporal Data. In Proceedings of the 2019 IEEE Conference on Games (CoG), London, UK, 20–23 August 2019; pp. 1–2.
7. Wu, Y.; Xie, X.; Wang, J.; Deng, D.; Liang, H.; Zhang, H.; Cheng, S.; Chen, W. Forvizor: Visualizing spatio-temporal team formations in soccer. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 65–75. [[CrossRef](#)] [[PubMed](#)]
8. Khan, A.; Lazzarini, B.; Calabrese, G.; Serafini, L. Soccer event detection. In Proceedings of the 4th International Conference on Image Processing and Pattern Recognition (IPPR 2018), Copenhagen, Denmark, 28–29 April 2018; pp. 119–129.
9. Russo, M.A.; Kurnianggoro, L.; Jo, K. Classification of sports videos with combination of deep learning models and transfer learning. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 1–5.

10. Opta Sports. F24: Feed Specification Document. Available online: <https://www.docdroid.net/ymMGPRQ/opta-playground-f24-documentation-pdf> (accessed on 13 October 2020).
11. Opta Sports. Homepage. Available online: <https://www.optasports.com/> (accessed on 13 October 2020).
12. Strong, M. How Opta Produces and Develops Its Unique Data. Available online: <http://web.archive.org/web/20160117072238/https://www.sportstradingnetwork.com/opta-produces-develops-unique-data/> (accessed on 13 October 2020).
13. Marlin-Bennett, R. *Knowledge Power: Intellectual Property, Information, and Privacy*; Lynne Rienner Publishers: Boulder, CO, USA, 2004.
14. McCann, Z. Player Tracking Transforming NBA Analytics. Available online: https://www.espn.com/blog/playbook/tech/post/_/id/492/492 (accessed on 13 October 2020).
15. Pappalardo, L.; Cintia, P.; Rossi, A.; Massucco, E.; Ferragina, P.; Pedreschi, D.; Giannotti, F. A public data set of spatio-temporal match events in soccer competitions. *Sci. Data* **2019**, *6*, 1–15. [[CrossRef](#)] [[PubMed](#)]
16. Richly, K.; Bothe, M.; Rohloff, T.; Schwarz, C. Recognizing compound events in spatio-temporal football data. In Proceedings of the International Conference on Internet of Things and Big Data, Rome, Italy, 23–25 April 2016; Volume 2, pp. 27–35.
17. Gudmundsson, J.; Horton, M. Spatio-temporal analysis of team sports. *ACM Comput. Surv. CSUR* **2017**, *50*, 1–34. [[CrossRef](#)]
18. Horton, M.; Gudmundsson, J.; Chawla, S.; Estephan, J. Automated classification of passing in football. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Ho Chi Minh, Vietnam, 19–22 May 2015; pp. 319–330.
19. Beetz, M.; von Hoyningen-Huene, N.; Kirchlechner, B.; Gedikli, S.; Siles, F.; Durus, M.; Lames, M. Aspogamo: Automated sports game analysis models. *Int. J. Comput. Sci. Sport* **2009**, *8*, 1–21.
20. Yue, Y.; Lucey, P.; Carr, P.; Bialkowski, A.; Matthews, I. Learning fine-grained spatial models for dynamic sports play prediction. In Proceedings of the 2014 IEEE International Conference on Data Mining, Shenzhen, China, 14–17 December 2014; pp. 670–679.
21. Wei, X.; Lucey, P.; Vidas, S.; Morgan, S.; Sridharan, S. Forecasting events using an augmented hidden conditional random field. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; pp. 569–582.
22. Richly, K.; Moritz, F.; Schwarz, C. Utilizing artificial neural networks to detect compound events in spatio-temporal soccer data. In Proceedings of the 2017 SIGKDD Workshop MiLeTS, Halifax, NS, Canada, 13–17 August 2017.
23. Morra, L.; Manigrasso, F.; Canto, G.; Gianfrate, C.; Guarino, E.; Lamberti, F. Slicing and dicing soccer: Automatic detection of complex events from spatio-temporal data. In Proceedings of the 17th International Conference on Image Analysis and Recognition (ICIAR 2020), Virtual Conference, 24–26 June 2020.
24. Schuiling, B. Gameplay Football. Available online: <https://github.com/BazkieBumpercar/GameplayFootball> (accessed on 13 October 2020).
25. Khaustov, V.; Mozgovoy, M. Learning Believable Player Movement Patterns from Human Data in a Soccer Game. In Proceedings of the 2020 22nd International Conference on Advanced Communication Technology (ICACT), Pyeong Chang, Korea, 16–19 February 2020; pp. 91–93.
26. Perš, J.; Bon, M.; Kovacic, S. Errors and mistakes in automated player tracking. In Proceedings of Sixth Computer Vision Winter Workshop, Bled, Slovenia, 7–9 February 2001; pp. 25–36.
27. Perš, J.; Bon, M.; Kovačič, S.; Šibila, M.; Dežman, B. Observation and analysis of large-scale human motion. *Hum. Mov. Sci.* **2002**, *21*, 295–311. [[CrossRef](#)]
28. Linke, D.; Link, D.; Lames, M. Football-specific validity of TRACAB’s optical video tracking systems. *PLoS ONE* **2020**, *15*, e0230179. [[CrossRef](#)] [[PubMed](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).