



From Movement to Events: Improving Soccer Match Annotations

Manuel Stein^(✉), Daniel Seebacher, Tassilo Karge, Tom Polk,
Michael Grossniklaus, and Daniel A. Keim

University of Konstanz, Konstanz, Germany
{manuel.stein,daniel.seebacher,tassilo.karge,
tom.polk,michael.grossniklaus,daniel.keim}@uni-konstanz.de

Abstract. Match analysis has become an important task in everyday work at professional soccer clubs in order to improve team performance. Video analysts regularly spend up to several days analyzing and summarizing matches based on tracked and annotated match data. Although there already exists extensive capabilities to track the movement of players and the ball from multimedia data sources such as video recordings, there is no capability to sufficiently detect dynamic and complex events within these data. As a consequence, analysts have to rely on manually created annotations, which are very time-consuming and expensive to create. We propose a novel method for the semi-automatic definition and detection of events based entirely on movement data of players and the ball. Incorporating Allen's interval algebra into a visual analytics system, we enable analysts to visually define as well as search for complex, hierarchical events. We demonstrate the usefulness of our approach by quantitatively comparing our automatically detected events with manually annotated events from a professional data provider as well as several expert interviews. The results of our evaluation show that the required annotation time for complete matches by using our system can be reduced to a few seconds while achieving a similar level of performance.

Keywords: Visual analytics · Sport analytics · Event analysis

1 Introduction

In numerous invasive team sports such as soccer, automatic video analysis is increasingly being deployed to collect spatio-temporal data, consisting of player and ball movement [10]. This data is collected to gain deeper insights into the respective sport in order to increase the efficiency of players, analyze opposing teams and, consequently, improve training and team performance. Without further processing and analysis, however, this data alone does not provide deeper insights into a match. In order to take full advantage of the data, analyses must be carried out and visualizations must be generated so that analysts can process the large amounts of retrieved movement data.

Companies such as Stats and Opta manually annotate basic events, such as passes, ball possession times, and fouls or penalties, as well as more complex events such as offside determination. Here, analysts manually inspect and annotate vast amounts of multimedia data, mostly many hours of video, based on predefined criteria. In addition to being extremely time-consuming and expensive, this manual annotation is also susceptible to human error, thereby reducing the quality of the event data. Existing approaches to automate event detection are divided into two fields: Automatic video analysis and direct analysis based on previously recorded movement data in combination with existing, basic events. Automatic video analysis has been used, for example, to try to detect events in television broadcasts or video recordings of soccer matches [3, 5, 9, 11, 12, 15–17]. Using these approaches, simple events such as corner balls or goals, can be easily detected. However, recognizing new kinds of events requires a considerable amount of effort, since separate algorithms have to be created for each event type. Furthermore, many algorithms rely on implicitly annotated data, such as the organizer’s logo appearing before replays or specific camera movements, to detect certain events such as corner kicks or penalties. These techniques can typically only be applied in a narrow set of circumstances and do not represent a robust, generalizable approach. Several systems also use movement data directly in order to recognize events [6, 7, 13, 18]. However, these concepts do not allow interactive definition of events and are typically designed for a narrow set of purposes, such as commenting on soccer matches. Consequently, de Sousa Júnior et al. [4] suggest the recognition of events in matches should be the core focus of current research on soccer analysis. The recognition of events based on underlying event patterns is mentioned as an example. In addition to automatically recording player and ball movement data at a reasonable spatio-temporal resolution, it is also necessary to automate event recognition in order to make the detection of events more cost-effective, faster, and more reliable. Automatic analysis can also reveal aspects that would not be found with manual analysis due to bias, human error, lack of time, or simple lack of knowledge. At the same time, intuition and expert knowledge are required to formulate a meaningful objective and to define event patterns that are interesting or important for the respective question, analyst, player, or team. Some event-types in soccer do not have a universally shared definition, but are instead defined differently depending on each team, and therefore cannot be recognized correctly without manual interaction.

In this paper, we propose an automated event detection system based solely on soccer movement data. Simultaneously, we enable analysts to efficiently and flexibly incorporate their intuitions in order to find complex patterns in events. The developed system gives the user the possibility to custom-define spatio-temporal patterns and then automatically search for them. In contrast to existing approaches, it allows expert knowledge and human intuition to be integrated into the automated analysis process. By directly using movement data to generate events, it closes the gap between raw multimedia data and event data and compensates the lack of flexibility and scalability of manual data annotation.

With the help of a visualization displaying identified events in combination with the associated movement data, the presented system allows the correction of existing event patterns, the creation of new patterns, and the general assessment of event data. Since the recognition of events is automated, the system therefore supports real-time analysis.

2 Detection of Complex Events

The processing of complex events is a broad field in computer science that includes not only their processing, but also the architecture for processing data or event streams and the recognition of (complex) events. The most important step in event processing consists of detecting *complex events* and is therefore the main focus of our proposed approach. The detection of complex events in large soccer data enables analysts to summarize the increasing amounts of gathered movement data by highlighting interesting occurrences. This helps soccer analysts and coaches interpret the data faster and more effectively, giving them more time to use the resulting knowledge. In the following, we define an *event* as a time interval or point containing associated objects and attributes. For example, a shot on goal event can be associated with the players specifically involved. Furthermore, we define *event types* as events with certain common characteristics. An *event type* has a name and contains a definition of the characteristics common to its events, enabling the user to identify all associated events. A constellation of event types is called a *temporal pattern* or an *event pattern*. A sequence of events that satisfies the event pattern is an instance of a complex event type, also called a *complex event*. Figure 1(a) shows an example event pattern involving the event types *B*, *A* and *C*. In the example shown, event type *A* is starting before *B* is fully finished. The identified locations of the user defined event pattern in the overall event data stream are highlighted in Fig. 1(b). Complex event types themselves can occur in event patterns, resulting in a hierarchy of event patterns as shown in Fig. 2. For a better understanding, all subsequent steps are explained using the example of the offside rule, since the rules for this event are generally known and many technical details of event detection can be explained using them.

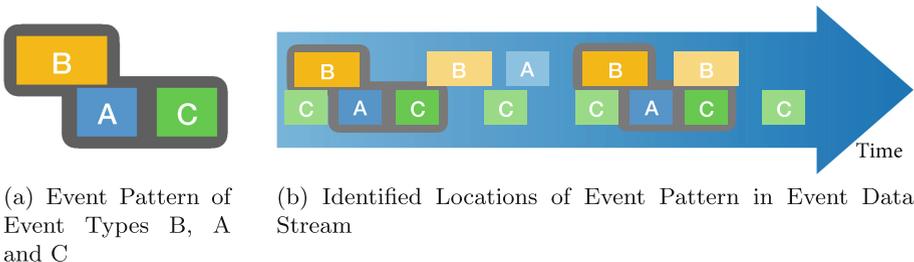


Fig. 1. Event pattern and event data stream with pattern locations

The Laws of the Game of the International Football Association Board state that a player is in an offside position if a player is in the opponent’s half of the pitch and closer to the goal line than both the ball and the second-to-last opponent [2]. In this description, we can see that several event types play a role in the offside rule and that they occur within a certain time constellation. Together, the event types form a complex event pattern including, for example, players and teams. Examples of involved event types in the offside rule are *player is in offside position* or *player is touching the ball*. The description of an event in natural language must be formalized in order to be usable for a program to find the event. One very similar and, therefore, intuitive formalism for describing temporal relationships of natural languages is given in *Allen’s interval algebra* [1]. In this algebra, possible topological relations (*qualitative relations*) can be expressed between intervals which, in our case, translate to events. For example, the relationship between two events can describe whether both events happen at the same time or directly one after another.

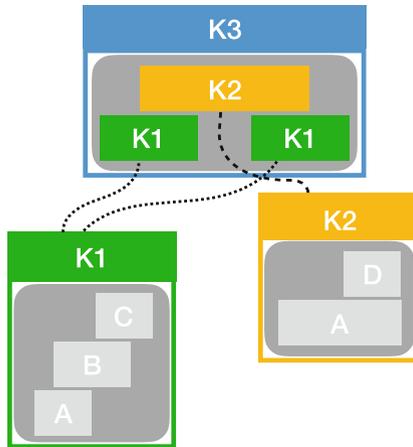


Fig. 2. Complex event patterns can be hierarchically composed by combining primitive or other complex events.

Some soccer events are not perceived as a time interval, but rather as a point in time. For example, the offside rule speaks about the (...) *point in time at which a ball is touched by a fellow team member (...)*. In order to use specific points in time, we need to extend Allen’s interval algebra. However, these changes are not drastic, since a point in time can be understood as an interval whose start and end points are the same. There are some special cases that need to be considered, when including time points into Allen’s interval algebra. For example, if A is an event that occurs at a point in time and B an event that has a time interval, not all relations can occur between these events, such as *DURING*, since $A_{start} = A_{end}$. An overview over all possible relations between time points

and intervals is given in Fig. 3 and an overview of all possible relations between two time points in Fig. 4.

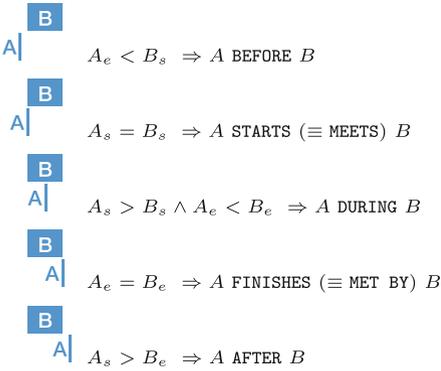


Fig. 3. Possible relations between a time point and a time interval

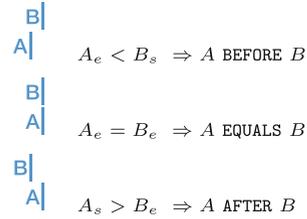


Fig. 4. Possible relations between two time points

These relations between the events allow us to create complex event patterns, as is the case, for example, with an offside. The example in Fig. 1(a) can serve as an illustration. B is the event “player plays the ball”, which must overlap with or at least meet event A “player is in an offside position”. This can be followed by another event C such as “Player A shoots at the goal”. These events, such as “player is in the offside” position, can be hierarchically composed by small events, as shown in Fig. 2. In addition, multiple conditions can sometimes apply to a particular event pattern, as described by the IFAB laws of the game. For example, the player can be in an offside position before the ball is played, but also when both events start at the same time. In addition, multiple conditions may apply to a particular event. For example, the player can be in an offside position before the ball is played to him, but also if both events start at the same time. To create and visualize such ambiguous event patterns, we introduce the concept of whiskers. Whiskers are a graphical addition to the well-known rectangular representation of intervals, which makes it possible to model ambiguous relationships between intervals. In the example in Fig. 5(a) we use these whiskers to model that event A can start simultaneously with event B, but A must start before B is finished at the latest. This ambiguous relationship could not be represented by simple rectangles.

However, defining these patterns is only the first step in making the transaction data useful. These patterns must also be found in the data. This process can be very time-consuming, which is why it makes sense to check beforehand whether a sample can be found at all. An example of such an undetectable pattern of three intervals (I, J, K) is ($I \text{ BEFORE } J, J \text{ BEFORE } K, K \text{ BEFORE } I$).

With the help of the path consistency algorithm, such relations can be checked for consistency in polynomial time.

After checking the consistency of the patterns, they can be searched for in the dataset. For pattern identification, we proceed similarly to Kempe [8], which uses deterministic finite automata as pattern identifiers. The complete process is outlined in Fig. 5. We want to find the offside pattern from Fig. 1(a) in the set of events which are sorted by time. For this, we start with an empty pattern identifier (a). The first matching event is B1. We duplicate the pattern identifier and insert the event B1 (b). The same procedure applies to the remaining events. After several steps, we have processed the last event C3 and see that three pattern identifiers are complete and thus the complex event offside was found three times in our data set.

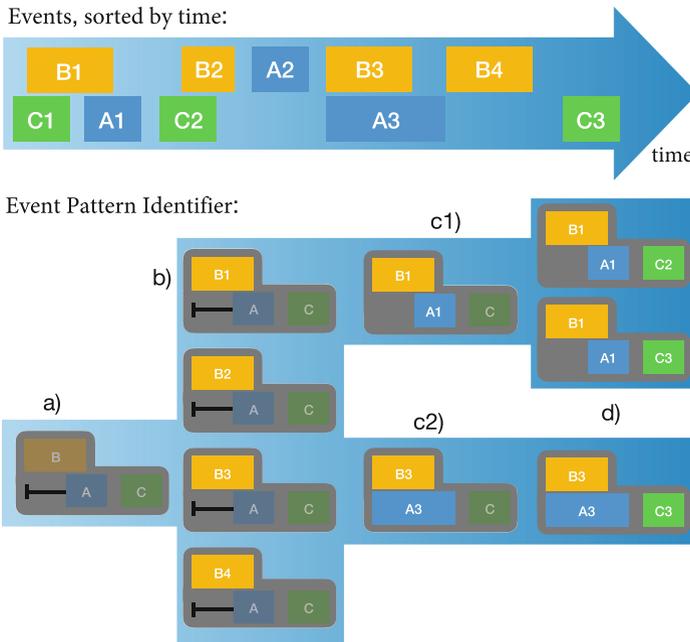


Fig. 5. Process of event detection: the starting point is an empty pattern identifier (a). The first event matching the pattern identifier is B1, the pattern identifier is duplicated and B1 is inserted into one of the duplicates (b). C3 is the last event and matches two pattern identifiers (c1, c2), which are therefore both duplicated. At the end, three pattern identifiers are complete, so the complex event was found three times (d).

3 System

We introduce a visual analytics system in order to define, search, and save events in soccer, based on the movement data of players and the ball. An overview about

the system can be seen in Fig. 6. Based on the available position data, soccer-specific properties such as the speed of the players or their distance to the ball are calculated to generate the basic events. This allows the detection of instances of the event type *Player is less than 1 m away from the ball*. Basic event types can be graphically arranged by the user in the system to build event patterns and, therefore, define a complex event. The system then automatically interprets the graphically displayed patterns as expressions of Allen’s interval algebra, which allows qualitative statements about the (temporal) order of events (like *Event A is before event B* or *Event A contains event B*). Created event patterns are searched in the data and the results of the search are visualized on a timeline as well as an abstract soccer pitch. The timeline allows the user to draw conclusions about when a pattern occurred in a match, and whether the definition of the pattern was correct or whether it needs to be adjusted. The visualization on the soccer pitch helps analysts to understand the movements of the players and the ball during identified events.

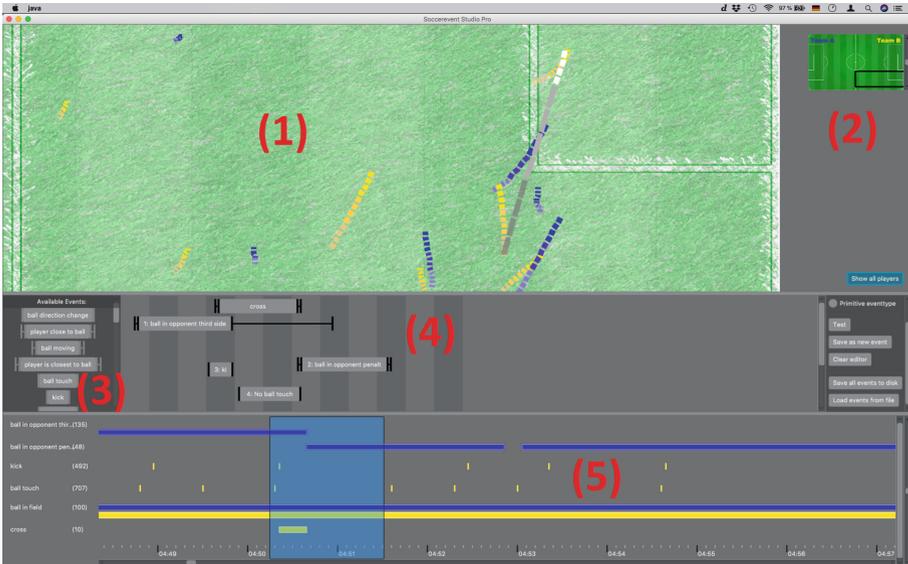


Fig. 6. System interface with abstract soccer pitch (1), mini map (2), library of events (3), event definition timeline (4) and timeline for displaying the results (5).

3.1 Defining and Detecting Basic Events

The starting point for the system is the movement data of the player and the ball during a soccer match. This data is not yet event data, unless we consider any recorded point as an event which, however, would not be practical. In our case, the movement data consist of two-dimensional coordinates for every tenth of a second of a match for every player and for the ball. The basic events are

generated assuming that the available movement data is continuous, i.e., in a straight line or a slight curve from one recorded coordinate to the next. All implemented soccer-specific properties for basic events are listed in Table 1. The calculation of most features is straight-forward, such as the distance between a player and the ball and the features derived from it, such as the closest player to the ball. For features that have a temporal aspect, such as speed, we use linear interpolation between the individual points in time to calculate these features. The deflection of the ball, however, is more demanding. Here we use the algorithm of Visvalingam [14], which removes points from the trajectory of the ball, which lie approximately on a straight line. The remaining points are those where the ball has made a sufficiently strong turn or changed direction. To calculate *offside position* and *closest to ball*, naive detection algorithms were implemented, which sort the players according to their distance to the goal or the ball.

Table 1. Used properties for generating basic events

| Property | Type | Generates an event, if... |
|--------------------------|----------|--|
| Speed | Interval | ...the speed has a certain value, or exceeds or falls below a certain threshold |
| Position | Interval | ...the position is within a certain range. Some special positions on the soccer pitch are predefined, otherwise one or more rectangular areas can be defined |
| Closest to ball | Interval | ...a player is at least as close to the ball as everyone else |
| Distance to ball | Interval | ...a player has a certain distance to the ball, exceeds or falls short of it |
| Distance between players | Interval | ...players have a certain distance to each other, exceed or fall short of each other |
| Offside position | Interval | ...a player is in an offside position |
| Deflecting the ball | Point | ...the ball changes direction |

Due to the basic event types, no great gain in knowledge is yet to be expected. Complex event types are required to show hidden connections or find interesting situations more quickly in a match. Our proposed system provides a list of already defined events, both basic and complex, displayed in our system as interactive rectangles. If an event is an interval, the rectangle always has whiskers, which enable the modeling of ambiguous interval relationships such as *BEFORE* \wedge *STARTS WITH*. If it is a point, it has no whiskers. Existing events can be positioned and arranged on the integrated visual timeline via *Drag and Drop*. This interaction concept is based on video editing software in which users arrange video and audio snippets in time to create a more complex end product. By dragging the sides of the rectangle, these can be widened or narrowed and, as a consequence, the relation of the involved event types to each other can be defined. To modify the remaining parts of the event pattern, relationships to other event types can be set in the timeline as well as minimum and maximum

duration time. Furthermore, it can be specified whether the new event type is to be an interval or a point, which either adds or removes the whiskers. After creating a new complex event, it is saved permanently in the event library. An example of a definition of a *cross* event can be seen in Fig. 7.

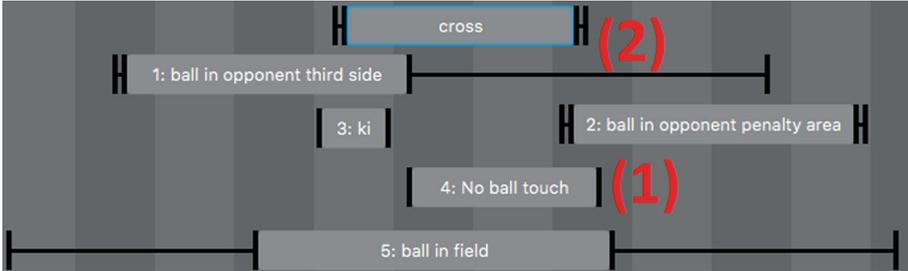


Fig. 7. Timeline enabling analysts to graphically define complex event types. All whiskers and rectangle sides in one of the strips are treated as if they had the same position (1). In the shown case, the reference interval is called “cross” (2).

3.2 Analyzing Identified Events

Complex events can be used both during their definition as well as afterwards to find and assess match situations in large amounts of soccer data. To provide an overview for the analyst, we visualize the locations of the identified event pattern in an additional, zoomable timeline, representing the entire match from kick-off to final whistle, as shown in Fig. 6(5). In this timeline, each interval is visualized by a rectangle, the left edge indicating where the event begins and the right edge of the rectangle indicating where the event ends. Since according to our event definition, point events are interval events where start and end time are the same, they are indicated by a rectangle with a very small width where the left edge is also at the point the event occurs. By giving the rectangles the respective team colors, we enable an efficient overview of the distribution of events and which team was responsible for which events at which time. The intervals or points displayed in the timeline are used, among other purposes, to quickly jump to match situations in which the event pattern occurs. In order to better assess the found complex events, the movement of the involved players and ball is visualized on an abstracted soccer pitch, when selecting a time period in the timeline. The trajectories are visualized as fragmented lines, as shown in Fig. 6(1). The color of a line corresponds to the team color. We use an additional color gradient on each trajectory, going from transparent at the start of the trajectory to opaque at the end, to indicate the direction of movement, without the need for animation. Additionally, we use the distance of the fragments in the line to indicate the speed of the players involved and the ball. In slower sections, the fragments are closer together. To see more precisely how player and ball trajectories progress or in order to get an overview, the abstracted soccer pitch can be zoomed in or out

by scrolling and moved by clicking and dragging. When zooming, the displayed area of the soccer pitch is drawn as a rectangle on an additional mini-map to improve the user’s orientation.

4 Evaluation

The goal of the evaluation is to show that most events, which are currently manually annotated, can be effectively defined and found with the system presented in this paper. Therefore, manually created event data sets of several matches are compared with their corresponding instances of event types defined in our system. The annotated event data has been collected by an established analysis company and consists of 43 manually annotated matches. First experiments have shown that the annotated events can partially lack accuracy due to human error and there is no detailed information available about the applied rules for annotation. For cross events, for example, it is not clear whether only successful crosses get annotated or an attempted cross is already enough to be annotated. In our definition for the automatic detection of cross events, we limit ourselves to only successful crosses. For quantitative analysis, statistical measures for the evaluation of proposed events are calculated. Afterwards, the differences between the data found and the manually generated data for individual cases are examined in more detail. To assess our method, we calculate the true positives (**TP**) (events found both manually as well as automatically), false positives (**FP**) (events that we found, but were not in the annotated data) as well as false negatives (**FN**) (manually annotated events that we did not find). The results of our quantitative evaluation, as displayed in Table 2, are very promising. All defined event types that are available both in the manually annotated datasets as well as our proposed system performed reasonably well, especially, considering that the rules

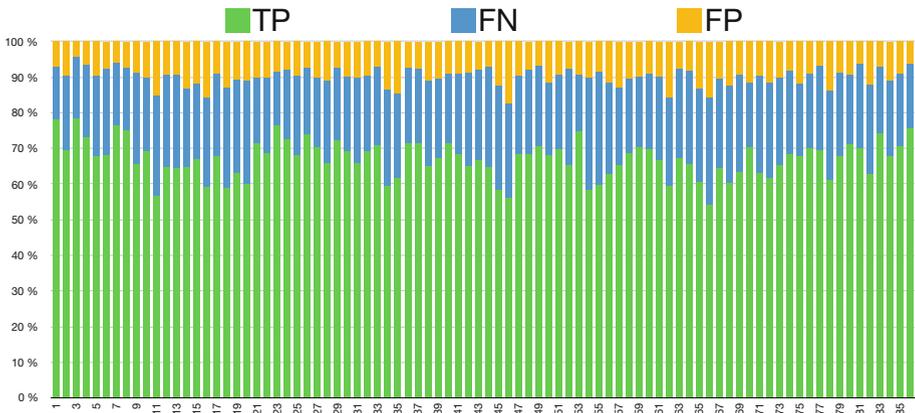


Fig. 8. Occurrences of true positives, false positives and false negatives for more than 36,000 passes in both half times of 43 matches (**P**recision = 87.5%, **R**eCALL = 75%, **F**-Measure = 80%)

and standards for manual annotation are not publicly available and might differ. Overall, the quantitative results of our proposed method demonstrate a good performance in detecting otherwise manually annotated events (Fig. 8).

Table 2. We assess our method quantitatively by comparing automatically detected events with manually annotated events from a professional data provider.

| Event | No. of events | Precision | Recall | F-measure |
|-----------------------------|---------------|-----------|--------|-----------|
| Passing | 36524 | 87.5% | 75% | 80% |
| Running with ball/dribbling | 23874 | 68% | 84% | 75% |
| Ball out of the pitch | 3012 | 78% | 98% | 87% |
| Goal | 129 | 97.5% | 91% | 94% |
| Cross | 1807 | 65% | 70% | 67% |
| Shot | 1124 | 82% | 49% | 62% |
| Reception | 28147 | 70% | 78% | 74% |

Furthermore, we evaluated the possibility of our system to define complex, hierarchic events such as *linebreaking passes* within several open interviews with two experienced soccer experts. The experts, one former coach from the youth department of the German soccer club FC Bayern München and one coach from the first team of an Austrian first league soccer club, confirm the potential of our proposed system. Both experts state they would make extensive use of such a system in order to reduce manual effort as well as be able to dynamically define own complex events.

5 Discussion and Conclusion

Our proposed system enables analysts to define and search for complex soccer events in large amounts of player and ball movement data. An important aspect that has not yet been included in this work is the inclusion of the context in the event definition, visualization, and subsequent analysis. Our system can already detect events such as crosses, but not why these events were carried out, for example, whether a player passed the ball to his teammate because he was under pressure, or because of a good free space. Enriching identified events with further context information can help analysts in order to define events more precisely and to have better information available for analysis. Another part that will be improved in future work is the visual analysis. Currently, we offer an overview of found complex events as well as a detailed view of individual events. A comparative view of events could help to identify common patterns and help analysts during match summarization. Our automatic event detection is also centered on events around the ball. We plan to offer more ways to efficiently integrate the definition of events to annotate defensive player behavior in our

system. Here, it would be interesting to define events when a player, for example, is blocking passing possibilities to other opposing players while attacking the ball possessing player. Another interesting set of events, according to our experts, would be to define complex events indicating when players are attacking in certain predefined pressing areas or when, for example, midfielders are too far away from the defenders of their team. Additionally annotating events when players are not close enough to opposing players would be helpful as well.

Finally, our work is an important step in the direction of automatic soccer analysis bridging the gap between movement data tracked by sensors or extracted from multimedia data and high-level analysis based on events. Our results are similar to manual event annotations but can be carried out in a fraction of the time and cost. Further improvements in the accuracy of event detection and the introduction of new events are planned for future work. Another main focus for subsequent research is not only on extending the detection of complex events but also on their assessment. For example, it should not only be detected that a pass has taken place, but also whether it was a good decision to pass to this specific player in the current state of the match. This would help soccer analysts and coaches analyzing large amounts of match data, efficiently making them aware of the relevant events for analysis and match preparation.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* **26**(11), 832–843 (1983)
2. International Football Association Board: Laws of the game (2018/2019). <http://theifab.com/document/laws-of-the-game>. Accessed 02 Aug 2018
3. Chen, M., Zhang, C., Chen, S.C.: Semantic event extraction using neural network ensembles, pp. 575–580. *IEEE*, September 2007. <https://doi.org/10.1109/ICSC.2007.75>
4. de Sousa Júnior, S.F., de Albuquerque Araújo, A., Menotti, D.: An overview of automatic event detection in soccer matches, pp. 31–38. *IEEE*, January 2011. <https://doi.org/10.1109/WACV.2011.5711480>
5. Ekin, A., Tekalp, A.M., Mehrotra, R.: Automatic soccer video analysis and summarization. *IEEE Trans. Image Process.* **12**(7), 796–807 (2003)
6. Gudmundsson, J., Wolle, T.: Towards automated football analysis: algorithms and data structures. In: *Proceedings of the 10th Australasian Conference on Mathematics and Computers in Sport*. Citeseer (2010)
7. Jensen, J.C.C.: Event detection in soccer using spatio-temporal data. Ph.D. thesis, Aarhus Universitet, Datalogisk Institut (2015)
8. Kempe, S.: Häufige Muster in zeitbezogenen Daten. Ph.D. thesis, Otto-von-Guericke University Magdeburg, Germany (2008). http://edoc.bibliothek.uni-halle.de/receive/HALCoRe_document_00005803
9. Kolekar, M.H., Palaniappan, K., Sengupta, S., Seetharaman, G.: Semantic concept mining based on hierarchical event detection for soccer video indexing. *J. Multimed.* **4**(5), 298–312 (2009). <https://doi.org/10.4304/jmm.4.5.298-312>
10. Stein, M., et al.: Bring it to the pitch: combining video and movement data to enhance team sport analysis. *IEEE Trans. Vis. Comput. Graph.* **24**(1), 13–22 (2018)

11. Wang, T., Li, J., Diao, Q., Hu, W., Zhang, Y., Dulong, C.: Semantic event detection using conditional random fields, p. 109. IEEE (2006). <https://doi.org/10.1109/CVPRW.2006.190>
12. Tavassolipour, M., Karimian, M., Kasaei, S.: Event detection and summarization in soccer videos using Bayesian network and copula. *IEEE Trans. Circ. Syst. Video Technol.* **24**(2), 291–304 (2014)
13. Tovinkere, V., Qian, R.: Detecting semantic events in soccer games: towards a complete solution, pp. 833–836. IEEE (2001). <https://doi.org/10.1109/ICME.2001.1237851>
14. Visvalingam, M., Whyatt, J.D.: Line generalisation by repeated elimination of points. *Cartogr. J.* **30**(1), 46–51 (1993)
15. Wickramaratna, K., Chen, M., Chen, S.-C., Shyu, M.-L.: Neural network based framework for goal event detection in soccer videos, pp. 21–28. IEEE (2005). <https://doi.org/10.1109/ISM.2005.83>
16. Tong, X.-F., Lu, H.-Q., Liu, Q.-S.: A three-layer event detection framework and its application in soccer video, pp. 1551–1554. IEEE (2004). <https://doi.org/10.1109/ICME.2004.1394543>
17. Yu, X., Xu, C., Leong, H.W., Tian, Q., Tang, Q., Wan, K.W.: Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, p. 10 (2003)
18. Zheng, M., Kudenko, D.: Automated event recognition for football commentary generation. *Int. J. Gaming Comput.-Mediat. Simul.* **2**(4), 67–84 (2010). <https://doi.org/10.4018/jgcms.2010100105>