StreamTeam-Football: Analyzing Football Matches in Real-Time on the Basis of Position Streams

Lukas Probst*, Heiko Schuldt*, Philipp Seidenschwarz*§, and Martin Rumo§
*Department of Mathematics and Computer Science, University of Basel, Switzerland
§Centre for Technologies in Sports and Medicine, Bern University of Applied Sciences, Switzerland
{lukas.probst, heiko.schuldt, philipp.seidenschwarz}@unibas.ch, martin.rumo@bfh.ch

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9377914

***Abstract*—In recent years, Big Data has become an important topic in many areas of our daily lives, including sports. Almost all professional clubs analyze matches to improve the performance of their teams. However, events are still predominantly captured manually, although many sensor-based and video-based tracking systems exist which provide the positions of the players and the ball in real-time. This manual process is tedious and error-prone. In this paper, we present STREAMTEAM-FOOTBALL, an open source football analysis application, that fills this gap. STREAMTEAM-FOOTBALL allows to analyze football matches fully automatically and in real-time on the basis of tracked position data using a data stream analysis approach. Our evaluation confirms the effectiveness of our automated analysis.**

***Index Terms*—Big Data Mining, Data Stream Analysis, Complex Event Detection, Team Sports Analysis, Football Analysis**

I. INTRODUCTION

In recent years, capturing and analyzing data became prevalent in team sports. Especially professional clubs gather large volumes of tracking data, events, and statistics that are produced with a high velocity and which contain a large variety of information. Clubs apply diverse techniques to analyze these Big Data in order to discover the strengths and weaknesses of their own and the opponents' teams.

But until today, matches in team sports are still not analyzed fully automatically. Although it is already common practice to generate statistics such as pass success rates by aggregating event data, the detection of events (e.g., passes and shots) and thus the generation of the event data for further analyses is still predominantly performed manually. On the one hand, there are commercial dataset providers such as Opta which sell event datasets which their employees capture manually [1]. On the other hand, there are companies which have brought software products to the market that enable match analysts of clubs, associations, or broadcasters to capture the events. For instance, Coach Capture [2], myDartfish Live S [3], and Sportscode [4] contain interfaces which assist users in capturing events manually while watching a live stream or a recorded video of the match. Moreover, there are approaches in academia to leverage cheap crowdsourcing platforms to generate event datasets. CrowdSport [5], for instance, aggregates events captured by multiple microworkers in video snippets into one event dataset. Despite their differences, these approaches have in common that human operators have to manually capture the events of a match either live during the match or post-match while reviewing videos of the match. This includes the selection of the correct event type as well as the assignment of the correct temporal and spatial information to each event.

This is despite the fact that there are diverse commercially available tracking systems, such as TRACAB Optical Tracking [6], which capture the position of the players and the ball. However, so far the positions captured by these systems are only used to complement manually captured events or to generate statistics, such as run distances and heatmaps, which do not require prior detection of events.

STREAMTEAM-FOOTBALL fills this gap by using the positions which are captured with one or multiple of these tracking systems to analyze football matches fully automatically and in real-time. Hence, STREAMTEAM-FOOTBALL does not require any manual interaction to detect events but derives them solely from the raw tracking data, i.e., from the player and ball positions. The code of STREAMTEAM-FOOTBALL is published under an open source license on GitHub.[1]

The contribution of this paper is threefold: We first present how STREAMTEAM-FOOTBALL performs the analyses in real-time, algorithmically, and stepwise in a data stream analysis system on the basis of a raw position stream (see Section II). Second, we present how STREAMTEAM-FOOTBALL provides coaches with live feedback and developers with an effective debugging tool by means of visualizing the analysis result in real-time (see Section III). Third, we evaluate the strengths of STREAMTEAM-FOOTBALL's analysis approach by assessing its analysis quality (see Section IV).

II. ANALYSIS

STREAMTEAM-FOOTBALL is a novel application which allows to analyze Big Data on football matches fully automatically and in real-time on the basis of tracked player and ball positions and additional match metadata such as the field dimensions. The analysis is performed stepwise in a workflow formed by multiple workers of a worker-based data stream analysis system, with a dedicated worker for each subtask of the overall analysis task.

This work was partly supported by the Hasler Foundation in the context of the project *StreamTeam: from Individual Sensing to Collaborative Action Analysis* (contract no. 16074).

[1] STREAMTEAM GitHub project: https://github.com/streamteam (ver. 1.0.1).

All worker implementations in STREAMTEAM-FOOTBALL follow an algorithmic approach with the objective to convert vaguely-defined terms like pressing which still lack a clear, universally accepted definition into clearly modeled concepts that can be converted into algorithms [7]. This has been done in close collaboration between computer scientists, sports scientists, and practitioners (coaches).

A. Worker-Based Data Stream Analysis

There are many diverse data stream analysis systems which differ in the way the overall analysis task is specified. As mentioned above, STREAMTEAM-FOOTBALL follows a worker-based data stream analysis approach. Representatives of this data stream analysis system category are systems like Apache Storm [8], Apache Samza [9], MillWheel [10], PAN [11], and OSIRIS-SE [12]. The fundamental idea of worker-based data stream analysis systems is to perform the analysis stepwise in a workflow consisting of multiple independent and freely programmable workers.

Each worker performs a subtask of the overall analysis. For doing so, every worker consumes elements from a configurable set of input streams. These input stream elements can be emitted by external devices (e.g., sensors or a video-based tracking system) and by other workers of the workflow. Whenever an element of one of its input streams is received, the worker performs its code and emits arbitrary (incl. zero) many output stream elements which contain new analysis results of the worker. Moreover, each worker can additionally have a timer to periodically execute its code – a feature which is very helpful to generate time window based statistics.

Splitting the code of the overall analysis task into smaller fragments (one for each subtask) is beneficial for multiple purposes such as sharing intermediate results, modifying existing analyses, and adding new analyses. The major advantage of worker-based data stream analysis systems is that they enforce a clean code separation instead of only supporting it. This is beneficial in our scenario since we aim to assist domain experts such as match analysts without a profound software engineering background in developing their own analyses.

B. Stream Categories

As mentioned above, each worker consumes input stream elements and emits output stream elements. In our work, we group the input streams whose elements a worker can consume and the output streams whose elements a worker can emit into four categories: Raw input streams, event streams, state streams, and statistics streams.

Elements of *raw input streams* embody different forms of raw input data for data stream analysis systems. That is (in contrast to the elements of event, state, and statistics streams) raw input stream elements are not generated by components of a data stream analysis system but originate from external devices (e.g., sensors or a video-based tracking system).

Elements of *event streams* represent a certain event, such as a successful pass or a duel, which was detected by a data stream analysis system. Event streams are further distinguished into *atomic* and *non-atomic* event streams. An element of an atomic event stream contains all information of an atomic event (e.g., a penalty box entry event or a successful pass event). In contrast, an element of a non-atomic event stream comprises updates of a non-atomic event (e.g., a duel event), i.e., a prolonged event which is detectable from an early stage on and for which it is beneficial to emit updates while the event takes place (e.g., for visualization purposes).

Elements of *state streams* contain a state, such as the current areas spanned by the players of a team or the current pressing intensity on the player in ball possession, which was calculated by a data stream analysis system.

Elements of *statistics streams* comprise statistical values, such as the run distance or the pass success rate of a player or team, which were generated by a data stream analysis system.

C. Football Analysis Workflow

The workers of STREAMTEAM-FOOTBALL form a workflow which analyzes football matches stepwise in real-time on the basis of simple match metadata and a continuous stream of player and ball positions. Overall, STREAMTEAM-FOOTBALL's analysis workflow detects diverse atomic events, such as kick-offs, ball possession changes, set plays, shots, passes, and even pass sequences. Moreover, also non-atomic dribblings, duels, and under pressure situations are detected. In addition, STREAMTEAM-FOOTBALL generates many statistics, such as heatmaps, ball possession statistics, and pass statistics, and calculates states, such as a virtual offside line and information about the areas which the teams span.

To generate these analysis results, STREAMTEAM-FOOTBALL's analysis workflow consists of 14 workers (listed in Table I) which together consume elements of two raw input streams – elements of a match metadata stream and a raw position stream – and emit elements of 19 atomic event streams, three non-atomic event streams, four state streams, and nine statistics streams. Fig. 1 depicts the workflow formed by all 14 workers. In the following, we will present two workers in more detail in order to give an impression of the algorithmic analysis procedure. A detailed description of the other twelve workers can be found in [13].

1) Field Object State Generation Worker: This worker is the first worker of the football analysis workflow. Its purpose is to transform raw position stream elements into unified field object state stream elements with additional information. The idea to unify the raw input data in the first worker is inspired by Herakles' data abstraction approach [14]. The field object state generation worker consumes match metadata stream elements and raw position stream elements. Whenever a raw position stream element which ships the current position of a field object (i.e., a player or the ball) is processed, all data are extracted from this element and enriched with the velocity of the field object that is calculated by leveraging the latest two positions and timestamps of the field object which have been stored in the local state. Subsequently, the position and the velocity are scaled to SI units and the field axes are mirrored, if necessary. Moreover, the object and group

TABLE I
STREAMTEAM-FOOTBALL'S WORKERS

Worker	Analysis subtask	Input streams	Output streams
Field object state gen.	Transforms raw position stream elements into unified field object state stream elements with additional information	Match metadata, raw position	Field object state
Kick-off detection	Detects kick-offs and informs which team plays in which direction	Field object state	Kick-off event
Time	Informs about the current match time in seconds	Field object state, kick-off event	Match time progress event
Area detection	Detects if a field object (i.e., a player or the ball) enters or leaves an area	Field object state, match metadata	Area event
Set play detection	Detects free kicks, corner kicks, goal kicks, penalties, and throw-ins and generates set play statistics	Area event, field object state, kick-off event	Corner kick event, free kick event, goal kick event, penalty event, set play statistics, throw-in event
Ball possession	Detects ball possession changes as well as duels and generates ball possession statistics	Area event, field object state, kick-off event, match metadata	Ball possession change event, ball possession statistics, duel event
Offside	Generates a virtual offside line	Ball possession change event, field object state, kick-off event	Offside line state
Pressing analysis	Calculates a pressing metric and detects under pressure situations	Ball possession change event, field object state	Pressing state, under pressure event
Kick detection	Detects kicks (i.e., if the ball has moved away from the player in ball possession)	Area event, ball possession change event, duel event, field object state, kick-off event, match metadata, under pressure event	Kick event
Pass & shot detection	Detects successful passes, interceptions, misplaced passes, clearances, goals, and shots off target and generates pass as well as shot statistics	Area event, ball possession change event, kick event, kick-off event	Clearance event, goal event, interception event, misplaced pass event, pass statistics, shot off target event, shot statistics, successful pass event
Pass comb. detection	Detects pass sequences as well as double passes and generates pass sequence statistics	Area event, clearance event, interception event, misplaced pass event, successful pass event	Double pass event, pass sequence event, pass sequence statistics
Dist. & speed analysis	Detects speed level changes as well as dribblings and generates distance, speed level, and dribbling statistics	Ball possession change event, field object state	Distance statistics, dribbling event, dribbling statistics, speed level change event, speed level statistics
Team area	Generates information about the areas which are spanned by the players of the teams	Field object state	Team area state
Heatmap	Generates individual player and team heatmaps	Field object state, match metadata	Heatmap statistics

identifiers are renamed using rename maps contained in the match metadata stream element for the match. Finally, a field object state stream element which ships all information about the current state of the field object is generated and emitted.

2) Pass and Shot Detection Worker: This worker detects successful passes, interceptions, misplaced passes, clearances, goals, and shots off target. Moreover, it generates pass and shot statistics. For this purpose, it consumes elements of the area event, the ball possession change event, the kick event, and the kick-off event stream. Whenever a new area event stream element which ships the information that the ball left the field at a certain region or a new ball possession change event stream element which does not ship the information that no player is in possession of the ball is processed, the contained information as well as the information stored for the last kick event are used to check if a pass or shot occurred. To qualify for a pass or shot, the last kick event must not be already the start of the last detected pass or shot and the timestamp difference between the kick event and the currently processed area or ball possession change event must be positive (i.e., the kick event must have happened first) but not too large. If these temporal checks are passed, a pass or shot occurred. In this case, the information who kicked the ball, the playing direction of the kicking team, the field zone in which the ball was kicked, whether the kicking player was attacked, and the information where the ball left the field (if an area event is processed) or where and by whom (same or different team as the kicking player) the ball was received (if a ball possession change event is processed) are used to determine which pass or shot event occurred. For instance, a successful pass is detected if the ball was kicked and received by players of the same team. In contrast, a shot off target is detected if the ball leaves the field close to the goal of the opposing team and the kicking player was not attacked in the defense third (making the shot a clearance). In any case, i.e., if any pass or shot event (e.g., a successful pass) was detected, a corresponding pass or shot event stream element (e.g., a successful pass event stream element) is generated and emitted. Moreover, the pass or shot statistics for the kicking player and his/her team are updated and emitted in new pass or shot statistics stream elements.

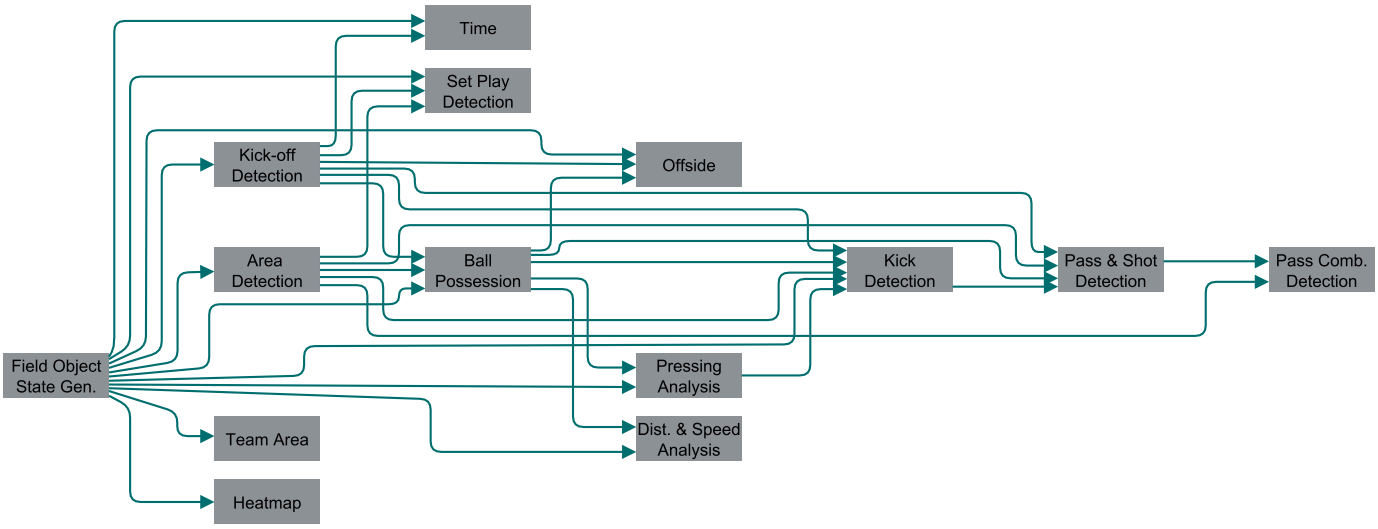

Fig. 1. StreamTeam-Football's Analysis Workflow. Each arrow visualizes that the worker at the arrowhead consumes elements of one or multiple data streams which the worker at the start of the arrow emits. Raw input streams in general and output streams which are not consumed by any worker are not depicted.

D. Implementation

STREAMTEAM-FOOTBALL is implemented as an analysis application on top of STREAMTEAM, our generic real-time data stream analysis infrastructure [13]. That is, its analysis workflow is implemented by specifying the code and the configuration of 14 STREAMTEAM workers.

The data stream analysis system at the heart of STREAMTEAM is based on Apache Samza [9] (ver. 0.13.1). In Samza, workers can be implemented as so-called jobs [15]. However, the workers of STREAMTEAM-FOOTBALL are not implemented directly using Samza's API. Instead, STREAMTEAM extends Samza's low-level API [16] (including its windowing feature [17]) by introducing an additional modularization level inside the workers that assist domain experts without a profound software engineering background (e.g., football match analysts) in implementing their own analysis workers with readable, well-structured, and duplicate-free code.

Moreover, STREAMTEAM improves Samza's support for key-based data parallelism. In Samza, jobs are split into tasks which perform the analysis subtask the job defines in parallel, each for a different partition of every input stream and thus for a different subset of the input data [15]. However, the splitting is done on a key-hash group instead of a key basis. Hence, each task processes input stream elements with different keys whose hash values belong to the same group. While this is a reasonable design choice from an architectural point of view, it has the pitfall that the worker developers have to take care of separating the analysis per key by themselves when implementing the code. To mitigate the development complexity for domain experts, STREAMTEAM adds additional extensions on top of Samza which facilitate implementing workers whose tasks perform the analysis strictly separated per key. As a result, STREAMTEAM-FOOTBALL is able to analyze multiple matches independently in parallel without an increased worker code complexity by setting the unique identifier of a match as the key of all data stream element which belong to the match.

III. VISUALIZATION

STREAMTEAM-FOOTBALL further visualizes the emitted analysis results in real-time. This visualization provides coaches and match analysts with live results during the match and proved to be an effective tool for football analysis worker developers to test if the workers produce the expected analysis results (e.g., if successful passes are detected correctly).

A. Features

STREAMTEAM-FOOTBALL's user interface provides many live visualizations which support coaches in making the right decisions during the match. Among others, the user interface visualizes detected atomic events and active non-atomic events on the field (see Fig. 2). Moreover, also heatmaps (see Fig. 3) and a virtual offside line (see Fig. 4) are visualized directly on the field. The other states and statistics are visualized as graphs and bar charts, respectively (see Fig. 5).

B. Implementation

STREAMTEAM-FOOTBALL's user interface is implemented as a platform-independent Web client. STREAMTEAM leverages Apache Kafka [18], a widely-used publish/subscribe system, for shipping data stream elements between components. However, the Web client is not able to directly interact with Kafka since there is no library for plain JavaScript which provides a Kafka consumer to pull data stream elements from the Kafka brokers. To solve this issue, the STREAMTEAM infrastructure is complemented by a Kafka REST proxy [13]. This component is a dedicated Kafka consumer which buffers the latest elements of all data streams and provides a REST API via which Web clients can access all data stream elements. STREAMTEAM-FOOTBALL's user interface uses this REST API to periodically pull the latest data stream elements that are required for the visualization for the currently selected match in data stream specific intervals.

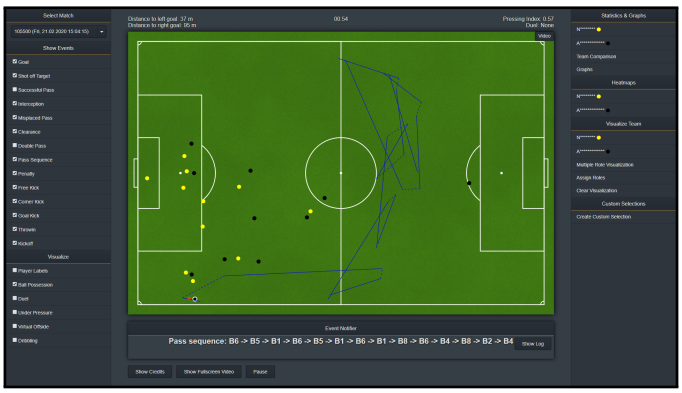

Fig. 2. Visualization of a Detected Pass Sequence Event. The solid blue lines illustrate successful passes and the dotted blue lines visualize the walked path between two passes of the sequence.

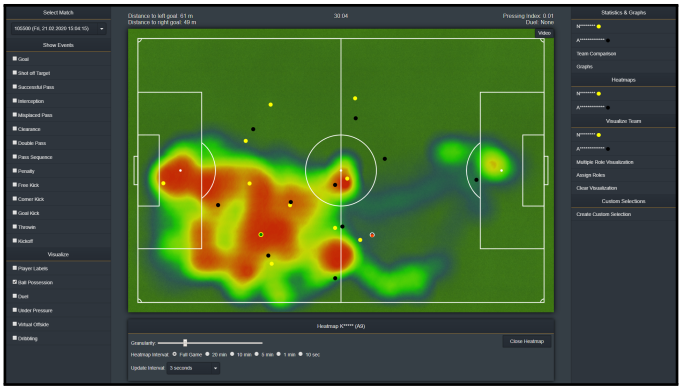

Fig. 3. Visualization of the Full Game Heatmap. The redder an area is the more often player A9 (highlighted in green) was positioned in this area.

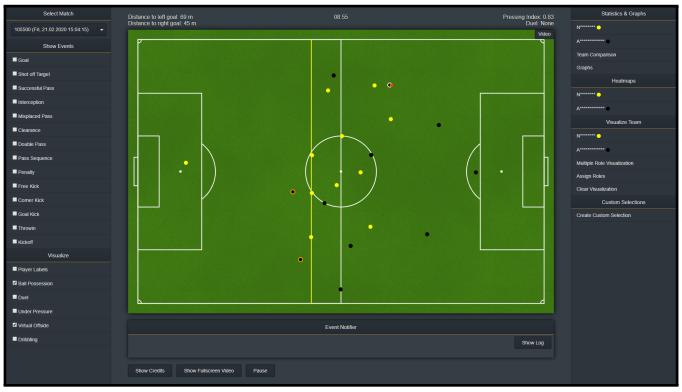

Fig. 4. Visualization of the Virtual Offside Line. The players who would be in offside position are highlighted with an orange border and the virtual offside line is illustrated with a yellow line.

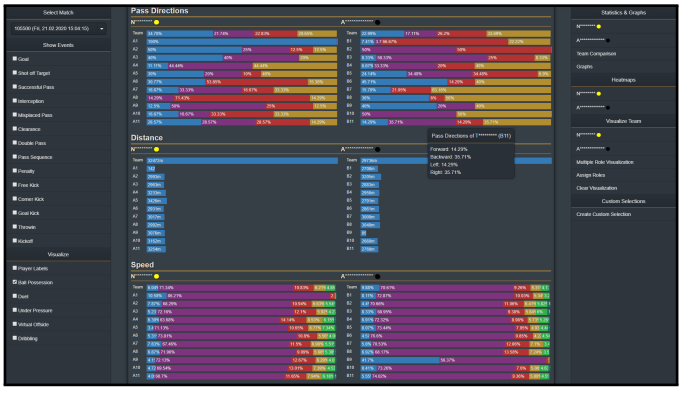

Fig. 5. Bar Charts of some Statistics. Each bar chart is updated in real-time and shows a tool tip when mouse hovered.

IV. EVALUATION

We have developed STREAMTEAM-FOOTBALL in close collaboration with sports scientists from the Swiss Federal Institute of Sport Magglingen (SFISM). That is, the sports scientists from the SFISM have provided us with definitions of events, states, and statistics which they have derived from interviews with football coaches [7] and which were the algorithmic basis of our implementations. In order to show that STREAMTEAM-FOOTBALL does not only aim for meaningful analysis results but produces correct ones, we evaluate the quality of STREAMTEAM-FOOTBALL's analysis results by comparing them to the Opta F24 data feed [19] of the same match. More precisely, we do so for the first halftime of a European top league match. Although Opta F24 data feeds are not perfect and can thus not be regarded as a ground truth, we use them for our evaluation since the Opta F24 data feeds are the de facto industry standard in football event datasets.

For our qualitative evaluation, only those analysis results of STREAMTEAM-FOOTBALL can be considered which are in some way also contained in the Opta F24 data feed. Thus, we concentrate on three atomic events, namely successful passes, interceptions, and throw-ins, which are detected by STREAMTEAM-FOOTBALL and contained in the Opta F24 data feed with similar temporal and spatial information.

The Opta F24 data feed is provided as an XML document. We have parsed this document and have extracted all successful passes, interceptions, and throw-in events from the first halftime. To obtain the successful pass, interception, and throw-in events which STREAMTEAM-FOOTBALL detects, we have generated the raw input stream elements of the same match by means of replaying a captured tracking dataset [13]. More precisely, we replayed the first halftime of the TRACAB Optical Tracking dataset [6] of the match. This dataset contains a new position for each player and the ball every 40 milliseconds. As discussed in [20], the positions are not completely free from errors but nevertheless exhibit a high quality.

To perform the evaluation, we compare the extracted events separately for each event type. For doing so, we iterate through the STREAMTEAM-FOOTBALL events in timestamp order and check if there is a matching Opta event. This is done by iterating through the Opta events in timestamp order and comparing each Opta event with the current STREAMTEAM-FOOTBALL event. Opta events which were already the match for a previous STREAMTEAM-FOOTBALL event with a lower timestamp are skipped in order to prevent that the same Opta event is used several times as the matching event. An Opta event matches a STREAMTEAM-FOOTBALL event if the difference between their timestamps is lower than a given time threshold and if the Euclidean distances between all positions which are extracted are lower than a given distance threshold.

	1 m	3 m	5 m	7 m	9 m	∞
1 s	0.00	0.05	0.12	0.15	0.15	0.39
2 s	0.01	0.27	0.51	0.58	0.59	0.74
3 s	0.01	0.33	0.63	0.71	0.73	0.87
4 s	0.01	0.33	0.64	0.73	0.75	0.90
5 s	0.01	0.33	0.64	0.73	0.75	0.92

(a) Correct Det. Pct. for Successful Pass Events

	1 m	3 m	5 m	7 m	9 m	∞
1 s	0.01	0.03	0.05	0.05	0.05	0.13
2 s	0.01	0.03	0.05	0.05	0.05	0.15
3 s	0.01	0.03	0.05	0.05	0.05	0.15
4 s	0.01	0.03	0.05	0.05	0.05	0.18
5 s	0.01	0.03	0.05	0.05	0.05	0.18

(b) Correct Det. Pct. for Interception Events

	1 m	3 m	5 m	7 m	9 m	∞
1 s	0.02	0.14	0.19	0.19	0.19	0.31
2 s	0.02	0.17	0.24	0.29	0.29	0.48
3 s	0.02	0.17	0.24	0.29	0.29	0.50
4 s	0.02	0.17	0.24	0.31	0.33	0.57
5 s	0.02	0.17	0.24	0.31	0.33	0.57

(c) Correct Det. Pct. for Throw-in Events

	1 m	3 m	5 m	7 m	9 m	∞
1 s	1.00	0.95	0.89	0.85	0.85	0.62
2 s	0.99	0.74	0.51	0.44	0.43	0.28
3 s	0.99	0.68	0.39	0.31	0.29	0.16
4 s	0.99	0.68	0.38	0.29	0.27	0.13
5 s	0.99	0.68	0.38	0.29	0.27	0.11

(d) Missed Det. Pct. for Successful Pass Events

	1 m	3 m	5 m	7 m	9 m	∞
1 s	0.95	0.91	0.82	0.82	0.82	0.55
2 s	0.95	0.91	0.82	0.82	0.82	0.45
3 s	0.95	0.91	0.82	0.82	0.82	0.45
4 s	0.95	0.91	0.82	0.82	0.82	0.36
5 s	0.95	0.91	0.82	0.82	0.82	0.36

(e) Missed Det. Pct. for Interception Events

	1 m	3 m	5 m	7 m	9 m	∞
1 s	0.96	0.77	0.69	0.69	0.69	0.50
2 s	0.96	0.73	0.62	0.54	0.54	0.23
3 s	0.96	0.73	0.62	0.54	0.54	0.19
4 s	0.96	0.73	0.62	0.50	0.46	0.08
5 s	0.96	0.73	0.62	0.50	0.46	0.08

(f) Missed Det. Pct. for Throw-in Events

Fig. 6. Qualitative Evaluation Results. (a)-(f) show the correct and missed detection percentages for the successful pass, interception, and throw-in events.

In order to investigate the spatial and temporal quality of the analysis results more thoroughly, we perform the event comparison not only once for each event type for a single time threshold and a single distance threshold but for multiple time and distance threshold combinations.

Fig. 6 shows the results of these comparisons. The percentages of correct detections in the upper row indicate the share of events detected by STREAMTEAM-FOOTBALL for which there is a matching event in the Opta F24 data feed. The missed detection percentages in the lower row indicate the share of extracted Opta events for which no matching STREAMTEAM-FOOTBALL event was detected.

As expected, the higher the thresholds are, the higher are the correct and the lower are the missed detection percentages. Moreover, the percentages reveal that the spatial accuracy has some issues while the temporal accuracy is acceptable. The fact that the correct detection percentages for the one meter distance threshold are almost zero for all event types independent of the selected time threshold shows that there are almost no STREAMTEAM-FOOTBALL events whose automatically derived position is very close to the manually assigned position of the corresponding Opta event. Moreover, all percentages are improved remarkably when changing the distance threshold from nine meters to ∞, hence when ignoring the spatial accuracy completely. In contrast, although increasing the time threshold from one to two seconds improves the percentages notably, further increasing the time threshold does not have the same effect as increasing the distance threshold. Due to the lack of an exact ground truth, the spatial inaccuracies cannot be fully explained. However, from reviewing the extracted events on a sample basis, we assume that at least some position mismatches are caused by the imperfect Opta F24 data feed. Furthermore, the evaluation results of CrowdSport show that microworkers have more problems in assigning correct spatial information than in assigning correct temporal information [5]. As the event capturing process in CrowdSport is similar to the one performed by the Opta employees [1], these results back our assumption that some spatial inaccuracies are not caused by STREAMTEAM-FOOTBALL but by imperfect positions in the Opta F24 data feed.

In addition, the numbers reveal that the detection quality depends on the complexity and well-definiteness of the events. The results for the successful passes show that STREAMTEAM-FOOTBALL can achieve a high detection quality if there is a clear, unambiguous, universally accepted event definition.

In contrast, the numbers show that STREAMTEAM-FOOTBALL exhibits a worse interception detection quality. One of the reasons for these results is the difference between the event definitions which are the basis of Opta's manual labeling and STREAMTEAM-FOOTBALL's algorithmic detection. Opta defines an interception as an intended pass which was intercepted by a player by moving into the pass and receiving or blocking the ball [21]. In contrast, STREAMTEAM-FOOTBALL regards all passes which are received by a player of the opposing team as interceptions as long as the ball was not kicked in the defense zone while the player was attacked. This is even the case if the pass was not intended and thus

for instance if the ball was not actually kicked but jumped away to a player of the opposing team after blocking a pass of this team. In consequence, due to the different definitions, even if STREAMTEAM-FOOTBALL's pass and shot detection worker performed exactly as intended, the set of interceptions which are extracted from the Opta F24 data feed would be only a subset of the set of interceptions which STREAMTEAM-FOOTBALL detects. Besides actual wrong detections (e.g., two interceptions are detected wrongly if a wrong ball hit is detected when the ball is very close to an opposing player during a successful pass) this is another reason for the fact that only 22 interceptions are extracted from the Opta F24 data feed but 79 interceptions are detected by STREAMTEAM-FOOTBALL and that the missed detection percentages are better than the correct detection percentages. This indicates that the detection quality can be very low if there are multiple different valid event definitions, especially when the detection quality is measured by comparing the detected events with events from another dataset that uses different event definitions.

Moreover, STREAMTEAM-FOOTBALL's throw-in detection quality is rather ambivalent. When taking a closer look, the numbers reveal that there are much less missed (down to $8\,\%$) than wrong (still $43\,\%$ with the highest thresholds) detections. We suppose that the main cause for wrong detections is the fact that STREAMTEAM-FOOTBALL detects a throw-in whenever the ball enters the field and no other set play (e.g., a corner kick) was recently detected. This approach works well if nothing unexpected happens. However, there are situations in which the ball leaves and re-enters the field without involving a throw-in or a corner kick, for instance if the ball enters the field behind the goal line, if the ball continues rolling after a foul, or if the player who should perform the throw-in changes. For a human, it is very easy to distinguish between such a corner case and an actual throw-in. However, covering all these corner cases is hard for an algorithmic approach, especially since throw-ins are not frequent enough to enable easy testing and the optimization of the detection algorithm. This shows that more complex events which engender the need to handle many corner cases can be problematic for the algorithmic detection approach followed in STREAMTEAM-FOOTBALL, especially if the frequency of these events is too low.

Nevertheless, despite of the imperfections in both STREAMTEAM-FOOTBALL and the reference Opta dataset, the results show that STREAMTEAM-FOOTBALL's analysis approach can be used in practice. Due to the modular worker-based approach of STREAMTEAM-FOOTBALL, worker implementations can be easily and individually adjusted to concrete event definitions supported by a coach or game analyst.

V. RELATED WORK

There are different approaches towards analyzing team sports matches automatically.

A. Video-Based Analysis

Firstly, there are academic approaches towards analyzing team sports matches which use the visual and/or aural features of a match video to detect events in the match and thus somehow mimic the human approach. For instance, Fleischman and Roy [22] leverage visual features (camera motions and scene categories), aural features (sound categories), and the closed captioning text in order to detect events in a baseball match using unsupervised learning methods. Moreover, Chen et al. [23] use visual features to detect events in an American football match by means of applying computer vision techniques.

B. Offline Position-Based Analysis

In addition, there are offline position-based analysis systems which analyze past matches on the basis of complete position datasets. More precisely, they consume a static dataset which contains all positions that were tracked during the match as the input for their analyses. In consequence, these systems have all benefits of static data analysis. That is, these systems have the option to iterate multiple times over the complete dataset and to access specific data items. Moreover, they are not bound to strict temporal requirements since the analysis is anyways not performed for a live match.

Typically, offline position-based team sports analysis systems leverage the plethora of well-established machine learning methods which have been developed for static datasets. For instance, Richly et al. [24], [25] detect events in position datasets of a football match by means of applying different machine learning approaches, namely classification with a Support Vector Machine (SVM), the k-Nearest Neighbors (kNN) algorithm, the Random Forest approach, and a three-layered Neural Network. Moreover, Sangüesa et al. [26] as well as Wang and Zemel [27] leverage machine learning approaches to classify plays in basketball matches. While Sangüesa et al. [26] apply a Principal Component Analysis (PCA) and diverse machine learning algorithms (classification trees, SVMs, and kNN), Wang and Zemel [27] leverage a normal Neural Network and a Recurrent Neural Network.

C. Real-Time Position-Based Analysis

STREAMTEAM-FOOTBALL and other real-time position-based analysis systems analyze live matches on the basis of continuous position streams. In consequence, these systems have all the disadvantages associated with data stream analysis, namely that they cannot access all input positions of the whole match from the beginning but see only volatile portions of the input data. Moreover, they have to perform the whole workload in real-time – which is at the same time a technical challenge and a major advantage since the analysis results are not only available after but even live during the match.

Earlier versions of STREAMTEAM-FOOTBALL have been presented in [7], [28], and [29]. While [28] presents the underlying vision, [29] focuses on how STREAMTEAM-FOOTBALL can be combined with SportSense [30]–[33], our team sports video retrieval system, to form a comprehensive and integrated team sports analysis infrastructure, and [7] focuses on the sports scientific aspects of our work.

First approaches [34]–[39] towards analyzing team sports matches in real-time on the basis of position streams were pro-

posed as solutions for the DEBS 2013 Grand Challenge [40]. The organizers of this challenge provided a RedFIR tracking [41] dataset of a football match and defined four analysis tasks, namely detecting shots on goal and generating running statistics, ball possession statistics, and heatmaps.

In addition, the Grand Challenge dataset has been used to demonstrate Herakles [14]. Herakles does not only solve the four analysis tasks of the Grand Challenge but detects for instance pass events and generates pass statistics. Moreover, Herakles has a remarkable user interface whose features are comparable to those which STREAMTEAM-FOOTBALL's user interface provides. However, Herakles performs much less analyses than STREAMTEAM-FOOTBALL. This is true in general as Herakles detects for instance no set play events, no dribbling events, and no pass sequence events. But the most significant difference is the fact that STREAMTEAM-FOOTBALL performs analyses which consider the interaction of multiple players or even whole teams – such as duels, pressing metrics, or team area states.

VI. CONCLUSION AND OUTLOOK

In this paper, we have presented STREAMTEAM-FOOTBALL, a comprehensive system to automatically analyze team sports Big Data on the basis of player and ball position streams. STREAMTEAM-FOOTBALL is the first system which performs complex team behavior analyses in a football match in real-time and visualizes the live analysis results in a user interface. It is implemented as an application on top of STREAMTEAM, our generic real-time data stream analysis infrastructure. Our evaluation has revealed the strengths (and also the limitations) of our worker-based algorithmic analysis approach.

In our future work, we aim at extending STREAMTEAM-FOOTBALL with new analyses by leveraging the collaboration with sport scientists and coaches. In the course of this, we plan to investigate the potential of applying Machine Learning techniques for analyses which are hard to capture algorithmically and to explore the potential of assigning events a certain probability [42] instead of detecting them in a binary way. Moreover, we plan to consider additional types of raw input data (e.g., physiological data), and adapt STREAMTEAM-FOOTBALL to analyze other team sports with different event sets, such as American football, ice hockey, and also eSports.

REFERENCES

[1] D. Nutz, "Opta Match Experience: Wie werden Daten und Statistiken beim Fußball erfasst?" https://www.goal.com/de/meldungen/opta-match-experience-wie-werden-daten-statistiken-fussball/1edqf26j61ajg1ni3kpe21r2mv, 2018.

[2] ChyronHego Corporation, "Coach Capture," https://chyronhego.com/products/broadcast-graphics/coach-capture/, 2020.

[3] Dartfish, "myDartfish Live S," http://www.dartfish.com/live_S, 2020.

[4] Hudl, "Sportscode," https://www.hudl.com/products/sportscode, 2020.

[5] F. Sulser *et al.*, "Crowd-based Semantic Event Detection and Video Annotation for Sports Videos," in *Proc. of CrowdMM*, 2014.

[6] ChyronHego, "TRACAB Optical Tracking," https://chyronhego.com/products/sports-tracking/tracab-optical-tracking/, 2020.

[7] P. Seidenschwarz *et al.*, "A Flexible Approach to Football Analytics: Assessment, Modeling and Implementation," in *Proc. of IACSS*, 2019.

[8] A. Toshniwal *et al.*, "Storm @Twitter," in *Proc. of SIGMOD*, 2014.

[9] S. A. Noghabi *et al.*, "Samza: Stateful Scalable Stream Processing at LinkedIn," *Proc. of VLDB Endowment*, vol. 10, no. 12, 2017.

[10] T. Akidau *et al.*, "MillWheel: Fault-Tolerant Stream Processing at Internet Scale," *Proc. of VLDB Endowment*, vol. 6, no. 11, 2013.

[11] L. Probst *et al.*, "PAN – Distributed Real-Time Complex Event Detection in Multiple Data Streams," in *Proc. of DAIS*, 2016.

[12] G. Brettlecker and H. Schuldt, "Reliable distributed data stream management in mobile environments," *Information Systems*, vol. 36, no. 3, 2011.

[13] L. Probst, "Spatio-Temporal Multi Data Stream Analysis with Applications in Team Sports," Ph.D. dissertation, University of Basel, 2020.

[14] T. Michelsen *et al.*, "Demo: Herakles – Real-time Sport Analysis using a Distributed Data Stream Management System," in *Proc. of DEBS*, 2015.

[15] Apache Samza Contributors, "Samza Documentation (ver 0.13) - Concepts," http://samza.apache.org/learn/documentation/0.13/introduction/concepts.html, 2017.

[16] ——, "Samza Documentation (ver. 0.13) - API Overview," http://samza.apache.org/learn/documentation/0.13/api/overview.html, 2017.

[17] ——, "Samza Documentation (ver. 0.13) - Windowing," http://samza.apache.org/learn/documentation/0.13/container/windowing.html, 2017.

[18] J. Kreps *et al.*, "Kafka: A distributed messaging system for log processing," in *Proc. of NetDB*, 2011.

[19] Opta Sports, "Opta data feeds," https://www.optasports.com/services/data-feeds/, 2020.

[20] E. Pons *et al.*, "A comparison of a GPS device and a multi-camera video technology during official soccer matches: Agreement between systems," *PLoS ONE*, vol. 14, no. 8, 2019.

[21] Opta Sports, "Opta's event definitions," https://www.optasports.com/news/opta-s-event-definitions/, 2018.

[22] M. Fleischman and D. Roy, "Unsupervised Content-Based Indexing of Sports Video," in *Proc. of MIR*, 2007.

[23] S. Chen *et al.*, "Play Type Recognition in Real-World Football Video," in *Proc. of WACV*, 2014.

[24] K. Richly *et al.*, "Recognizing Compound Events in Spatio-Temporal Football Data," in *Proceedings of IoTBD*, 2016.

[25] ——, "Utilizing Artificial Neural Networks to Detect Compound Events in Spatio-Temporal Soccer Data," in *Proc. of MiLeTS*, 2017.

[26] A. A. Sangüesa *et al.*, "Identifying Basketball Plays from Sensor Data; towards a Low-Cost Automatic Extraction of Advanced Statistics," in *Proc. of ICDMW*, 2017.

[27] K.-C. Wang and R. Zemel, "Classifying NBA Offensive Plays Using Neural Networks," in *Proc. of MIT Sloan Sports Analytics Conf.*, 2016.

[28] L. Probst *et al.*, "Demo: Real-Time Football Analysis with StreamTeam," in *Proc. of DEBS*, 2017.

[29] ——, "Integrated Real-Time Data Stream Analysis and Sketch-Based Video Retrieval in Team Sports," in *Proc. of Big Data*, 2018.

[30] ——, "SportSense: User Interface for Sketch-Based Spatio-Temporal Team Sports Video Scene Retrieval," in *Proc. of UISTDA*, 2018.

[31] P. Seidenschwarz *et al.*, "Combining Qualitative and Quantitative Analysis in Football with SportSense," in *Proc. of MMSports*, 2019.

[32] ——, "The SportSense User Interface for Holistic Tactical Performance Analysis in Football," in *Proc. of IUI Companion*, 2020.

[33] ——, "High-Level Tactical Performance Analysis with SportSense," in *Proc. of MMSports*, 2020.

[34] H.-A. Jacobsen *et al.*, "Grand Challenge: The Bluebay Soccer Monitoring Engine," in *Proceedings of DEBS*, 2013.

[35] Y. Wu *et al.*, "Grand Challenge: SPRINT Stream Processing Engine as a Solution," in *Proc. of DEBS*, 2013.

[36] M. Jergler *et al.*, "Grand Challenge: Real-time Soccer Analytics Leveraging Low-Latency Complex Event Processing," in *Proc. of DEBS*, 2013.

[37] K. G. S. Madsen *et al.*, "Grand Challenge: MapReduce-Style Processing of Fast Sensor Data," in *Proc. of DEBS*, 2013.

[38] A. Gal *et al.*, "Grand Challenge: The TechniBall System," in *Proc. of DEBS*, 2013.

[39] S. Badiozamany *et al.*, "Grand Challenge: Implementation by Frequently Emitting Parallel Windows and User-defined Aggregate Functions," in *Proc. of DEBS*, 2013.

[40] C. Mutschler *et al.*, "The DEBS 2013 Grand Challenge," in *Proc. of DEBS*, 2013.

[41] T. von der Grün *et al.*, "A Real-Time Tracking System for Football Match and Training Analysis," in *Microelectronic Systems*, 2011.

[42] E. Alevizos *et al.*, "Probabilistic Complex Event Recognition: A Survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, 2017.