Automated Football Examination: Algorithmic Processes And Data Frameworks

Omkar K Shinde[1], Neha Dinesh Shirsat[2]
[1, 2] Institute of distance and open learning University of Mumbai Vidyanagari,
Santacruz (E) , Mumbai-98, India
PCP- K.J Somaiya Institute of Technology,Mumbai

Abstract- *Examining a football match holds vital importance for coaches, teams, and players, and thanks to contemporary technology, an increasing amount of match-related data is being amassed. Firms now provide the capability to precisely and comprehensively track the positions of every player and the ball. Exploiting this positional data can yield substantial advantages. While certain companies offer fundamental analyses like statistics and basic inquiries, performing more advanced analyses remains a daunting task. In our investigation, we presume access solely to high-precision and detailed positional data for all players and the ball. This research paper introduces two tools. The first is a machine learning model designed to forecast the outcome of a football match based on positional data. The second tool is a visualization utility, employed for scrutinizing positional data and uncovering patterns and trends within it. In concert, these tools facilitate a more profound comprehension of a football match, enabling more informed decisions aimed at enhancing team performance.*

Keywords- Team-based athletics, positional information, path analysis, identifying events, grouping

I. INTRODUCTION

Recent progress in tracking technology has enabled the precise and detailed recording of the locations of mobile entities. Consequently, these systems are finding broader applications in diverse domains such as sports, urban development, military, location-based services, and animal studies. In the context of football, numerous companies already provide the capability to monitor the mobility of all players during a match and furnish rudimentary analytical resources.

Our project focuses on employing these technologies within team sports, particularly in the context of football, with a primary emphasis on the computer science aspect. With the data detailing player and ball movement, our objective is to perform a more sophisticated analysis that considers the interplays and associations between various trajectories, identifies team formations, and reveals overarching trends and patterns. Over the past few years, we have created several algorithms and utilities designed for trajectory analysis. This paper presents an overview of two of these tools.

Our initial tool, as described in Section 3, is algorithmically uncomplicated yet holds a pivotal role in sophisticated automated analysis. It utilizes the trajectories of both players and the ball to autonomously recognize fundamental occurrences within the football match, such as kickoffs, corner kicks, and passes. Our experimental findings affirm that our approach is effective and exhibits a notable level of precision for various event categories. Nonetheless, it's worth noting that certain event types pose challenges for automated detection when relying solely on positional data.

Our first tool leverages positional data to autonomously recognize a range of specific events that took place during the football match. These events encompass elements such as the match's kickoff, corner kicks, and passes. Our experiments demonstrated the efficiency and a notable level of precision of our method. However, it's essential to note that, despite the high level of accuracy achieved, completely eradicating errors in event detection based solely on positional data remains challenging.

Our second tool is dedicated to scrutinizing the movements of an individual player throughout the game, with a particular emphasis on recognizing repetitive movement patterns, known as sub-trajectory clusters. As an example, a left-wing attacker might repeatedly make runs from the center-line along the left side of the field toward the opposing team's goal. The primary objective of this tool is to automatically detect these repetitive patterns. Despite its demanding computational nature, our experiments have demonstrated that this approach adeptly identifies sub-trajectory clusters, which can subsequently be used for more in-depth analysis.

Our second tool, as detailed in Section 4, is centered around the analysis of an individual player's movements, yet it can be readily adapted for the analysis of multiple trajectories. We seek out repetitive movement patterns, which we refer to as sub-trajectory clusters. This approach is aimed at

addressing inquiries such as, "How is the ball transitioned from the defensive region to the offensive region?"We demonstrate a prototype that enables users to define precise cluster parameters, and this model consistently recognizes clusters in alignment with the provided criteria. Nevertheless, it's vital to emphasize that the present version of the prototype may not be well-suited for interactive utilization, as it demands a substantial duration to handle inquiries.

We consider our algorithms and implementations to be part of the emerging field of automated football analysis. Although this area of research is relatively recent, its importance and popularity are steadily increasing, as evidenced by the growing number of related studies. For instance, Kang, Hwang, and Li (2006) introduced a method to quantitatively assess the performance of football players. Their method relies on four distinct metrics, which consider various player regions and their kicking actions. Another approach, as proposed by Fujimura and Sugihara (2005), utilizes regions based on generalized Voronoi diagrams. Grunz, Memmert, and Perl (2009) also delve into the analysis of actions in football matches, but their focus encompasses broader actions compared to our fundamental events, and they employ different detection techniques.

II. PRELIMINARIES

The location of a mobile object can be expressed through its spatial coordinates, typically denoted as x, y, and possibly z, at a specific moment in time, denoted as t. Collectively, these parameters constitute a data point (t, x, y, z). When a sequence of these data points is organized in chronological sequence, it is termed a trajectory. A trajectory provides a visual representation of an object's motion.

Our investigation relies on data extracted from an actual football game, generously provided by ProZone, and subsequently anonymized. This dataset offers spatial granularity at the decimeterlevel and temporal granularity of at least 10 samples per second, encompassing the trajectories of all players. However, the data's exact accuracy remains uncertain. To facilitate our analysis, we require access to the ball's trajectory, which regrettably is not included in the dataset. Instead, the dataset contains a catalog of manually annotated match events, such as "touch" and "pass," observed by human observers during the game. We construct the ball's trajectory samples using the timestamps and geographic coordinates of each event. Nonetheless, given that these annotations have a resolution of just one meter and one second, the resulting ball trajectory is a mere approximation. We believe that more precise ball trajectory data could significantly enhance the outcomes detailed in Section 3.

We implemented both our algorithms and prototypes using the Java programming language, and our experiments were carried out on a typical PC equipped with an Intel dual-core processor clocked at 2.33 GHz and 2GB of RAM.

III. BASICEVENT DETECTION

Section A: Methods

In order to identify key events in a football game, we require access to both player and ball trajectories. Our event detection method operates on multiple event tiers. The foundational tier comprises physical events, which can be recognized without specific knowledge of football's rules but do require an understanding of the field's dimensions and markings. Examples of these basic events include "ball-out," "ball-in," and "touch." "Ball-out" and "ball-in" events occur when the ball exits or re-enters the field, and depending on their location, "ball-in" instances can be further categorized as "throw-in," "corner kick," and so on. A "touch" event is registered when the ball undergoes a change in speed or direction. These scenarios only necessitate access to the ball's trajectory.

Figure 1: Snapshot of the application for visualizing a match. The identified events are showcased in the section situated beneath the playing field.

We can enhance the accuracy of these events by considering the trajectories of the participants. For example, in the case of a "touch" event, we assume that the player nearest to the ball made the touch (although there can be exceptions). Similarly, when a "throw-in" happens, we attribute it to the player closest to the ball. This approach generates a list of progressively intricate events.

These more intricate events can be further refined in subsequent iterations. For instance, a "pass" event comprises two consecutive "touch" events performed by different players on the same team. The same principle applies to events like "possession," "pass intercepted," "shot on goal," and so on.

The highest level of events, encompassing occurrences such as "free-kick," "substitution," "offside," "foul," and "red-card," is achieved after yet another round of refinement.

Section B: Experimental Findings

The primary interface of the event-detection prototype is illustrated in Figure 1. It showcases an animated soccer field where a game is in progress. The events are presented in sync with the animation, situated in the section just below this playing field. A segment of the event list is exhibited in Figure 2 for reference.

34:51	Intercept	(37.6, -23.9, 0.0)	Player04	Player08
34:51	Touch	(38.9,-24.6, 0.0)	Player08	
34:53	Ball Out	(59.0, -26.0, 0.0)		
35:13	Corner Cross	(57.2, -29.1, 0.0)	Player23	
35:13	Touch	(57.2, -29.1, 0.0)	Player23	
35:13	Shot	(57.2, -29.1, 0.0)	Player23	
35:15	Touch	(54.3, -5.3, 0.0)	Player09	
35:15	Goalkeeper Catch	(54.3, -5.3, 0.0)	Player09	
35:15	Pass	(54.3, -5.3, 0.0)	Player09	14.1 m/s
35:16	Receive	(36.3, -2.0, 0.0)	Player17	

Figure 2: A selection from the catalog of identified events, providing details such as time, event type, coordinates, and the players involved, among other information.

Event type	*falsePositive*	*falseNegative*	F_1-score
touch	7	3	0.979
pass	2	2	0.955
intercept	1	1	0.960
ball-out	1	4	0.935
throw in	0	2	0.963
corner kick	0	1	0.968
goal kick	1	2	0.909
kick off	0	1	0.960
shot	1	3	0.875
goalkeeper catch	0	2	0.941
goal	0	1	0.960
foul	3	6	0.823
offside	5	3	0.556
free kick	0	3	0.953

Table 1: Overview of selected statistical outcomes for a subset of the identified events. A higher F1-score indicates better performance.

Our algorithm is capable of calculating all the events for an entire match within a matter of seconds, making execution time a negligible concern in practical applications. Consequently, our primary emphasis centers on the precision of the identified events.

In our testing, we employed data that incorporated a compilation of annotations, as previously noted. To gauge the accuracy of our event detection, we compare our event list to these annotations, assuming the annotations to be 100% accurate, even though this assumption may not necessarily hold true.

To quantify accuracy, we rely on the F1-score, which is determined as follows, and this metric serves as our measure for reporting accuracy.

$$\frac{2 \cdot \text{truePositive}}{2 \cdot \text{truePositive} + \text{falseNegative} + \text{falsePositive}}$$

During the comparison between the observed events and the annotations, we utilized the truePositive, falseNegative, and falsePositive metrics to calculate the F1-score. Table 1 presents the falsePositive and falseNegative values along with the corresponding F1-score for certain observed events. Notably, some occurrences like "penalty kick" or "red card" were excluded from the table because they did not occur during the game. Moreover, events like "yellow card" were not considered due to their inability to be detected without supplementary information.

The data presented in Table 1 suggests that automated detection of fundamental events is feasible. Several events can be identified with a notable degree of accuracy, particularly those at the foundational level. However, more advanced occurrences such as "foul" and "offside" cannot be reliably identified.

The precision of our results is influenced by various factors. We discovered errors in some of the trajectories as well as in the annotations themselves. Additionally, the ball's trajectory had to be reconstructed from the annotations, significantly affecting the identification of certain events. In general, it is challenging, if not impossible, to ascertain the accuracy of our event detection systems based on the available data. Considering that any further refinement would only enhance accuracy for the existing dataset, lacking the actual ball trajectory, we opted against further refining our algorithms

IV. SUB TRAJECTORY CLUSTERING

Section A: Methods

Our tool's methodologies draw inspiration from the research conducted by Buchin, Buchin, Gudmundsson, Löffler, and Luo in 2008. In the following section, we will provide an overview of these methodologies and present the outcomes of our experiments.

There exist multiple techniques for measuring the similarity between two trajectories, which include the Longest Common Subsequence model (Vlachos, Gunopulos, and Kollios, 2002), the summation of parallel, perpendicular, and angular distances (Lee, Han, and Whang, 2007), and the calculation of average Euclidean distances between paths (Nanni and Pedreschi, 2006). In our approach, we employ the Frechet distance, a metric designed for comparing the similarity of continuous structures like curves and surfaces. It is defined through reparameterization of the forms and is considered a more suitable similarity metric for curves when compared to the Hausdorff distance (Alt, Knauer, and Wenk, 2004).

The concept of the Frechet distance can be visualized by considering a scenario in which a person is walking their dog on a leash (see Figure 3). The dog follows a specific path, denoted as Td, while the person follows their own trajectory or path, represented as Tp. The Frechet distance between Tp and Td is the minimum leash length that allows both the person and the dog to traverse their respective paths. Importantly, they can adjust their speed or even pause during the walk, but they are not allowed to backtrack on their routes.

Figure 3: Depicting the leash distance between an individual and their dog as they traverse their respective paths

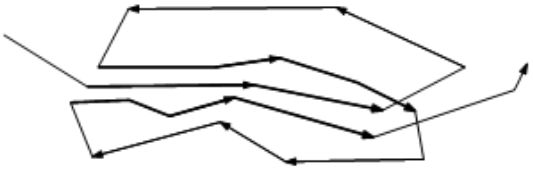

Figure 4: A subset of a trajectory displayed as a cluster (highlighted in bold).

The Frechet distance can be computed through two methods: the continuous and discrete approaches. In the continuous version, the person and their dog move smoothly along their paths, whereas in the discrete version, they progress from one vertex to the next on the path. We opted for the discrete Frechet distance, as it involves simpler implementation algorithms, as suggested by Buchin, Buchin, Gudmundsson, Löffler, and Luo in 2008.

During the course of a match, a player might traverse specific paths multiple times. Consequently, when examining a player's trajectory, denoted as T, certain segments of it may group together, forming a subtrajectory cluster, as illustrated in Figure 4. We adopt the approach pioneered by Buchin, Buchin, Gudmundsson, Löffler, and Luo in 2008, wherein a subtrajectory cluster is defined based on three parameters: m, l, and d. In this context, m signifies the presence of m distinct, non-overlapping subtrajectories within the trajectory T that constitute a subtrajectory cluster. The distance between these subtrajectories is limited to at most d, and at least one of these subtrajectories must have a length of l.

Precisely calculating subtrajectory clusters, as highlighted by Buchin, Buchin, Gudmundsson, Loffler, and Luo in 2008, presents a notably complex problem. Determining whether a given trajectory T possesses a subtrajectory cluster that meets the specified criteria of m, l, and d is an NP-hard challenge. Even when we attempt to approximate the values of m or l within certain factors and d within a factor of 2, the problem of maximizing either the number of subtrajectories or the length of subtrajectories (while keeping the other parameters fixed) remains NP-hard.

To provide some intuition, an NP-hard problem cannot be efficiently solved by any known algorithm. Consequently, we explore approximation strategies that roughly approximate d within a factor of 2.

The principal discovery from Buchin, Buchin, Gudmundsson, Loffler, and Luo in 2008 can be summarized as follows: When provided with a trajectory T, there exists an algorithm capable of computing a subtrajectory cluster with the maximum length, employing the discrete Frechet distance metric, with the distance d being approximated within a factor of 2 (meaning subtrajectories can have a distance twice as large as the specified parameter d). This method consumes O(nl) space and operates in O(n^2 + nml) time, where n represents the number of vertices in T, l signifies the maximum number of vertices within a subtrajectory, and m denotes the total count of subtrajectories within this cluster.

The operational prototype allows for configuration settings to dictate when subtrajectory clusters should be reported. These settings include criteria such as the acceptable distance (representing the maximum Frechet distance permissible between subtrajectories), the minimum cluster size, duration, and length. These criteria serve as thresholds to prevent the reporting of exceedingly small and insignificant clusters.

Section B: Experimental Outcomes

Our objective is to assess the efficiency and efficacy of our subtrajectory clustering algorithm through experimental assessments on the dataset. The purpose is twofold: first, to evaluate the algorithm's performance in terms of extracting valuable insights, and second, to gauge the algorithm's processing speed. We aim to ascertain whether the algorithm's runtime allows for the creation of an interactive tool wherein an analyst can define cluster parameters, and the tool promptly identifies the relevant clusters.

Section C: Usefulness

Figure 5: Images illustrating the trajectory of a left-wing player advancing and a subtrajectory cluster associated with it.

Figure 6: Images depicting the trajectory of a right-wing player moving in reverse and an associated subtrajectory cluster.

The utility of the clustering can be assessed through a visual examination of the results. This is achieved by displaying the clusters alongside the trajectory on an image of a football field. We provide illustrative snapshots that portray the entire trajectory for one half of the game, encompassing both the ball's path and the routes taken by two players (Buchin, Buchin, Gudmundsson, Loffler, and Luo, 2008). One specific cluster is highlighted in red and yellow, with the yellow subtrajectory serving as a representation of the cluster. Figure 5 presents the trajectory of a left-wing player (Player1), and the cluster reveals that numerous of his attacking movements in the first half of the game originated near the center of the left half of the centerline. These movements describe a long arc forward and toward the left side of the field before sharply turning back toward the center of the field.

Figure 7: Images displaying the trajectory of the ball and an associated subtrajectory cluster related to goal kicks.

The assessment of the practicality of the clustering process can be accomplished through a visual inspection of the outcomes, where the clusters are overlaid onto an image of a soccer field alongside the trajectory. The screenshots presented in the study demonstrate that the subtrajectory clustering effectively identifies clusters based on the defined parameters. Nevertheless, it is suggested that domain experts may find longer and larger clusters, comprising a greater

number of subtrajectories, more intriguing. To enhance the significance of the clustering results, it is advisable to consider a combination of geometry-based clustering and event-based clustering. Additionally, examining the trajectories from multiple matches may yield longer and more substantial clusters and unveil noteworthy emerging patterns. It's important to acknowledge that pursuing these recommendations may result in extended processing times.

Section D: Processing Times

The outcomes of our experiments suggest that the algorithm's processing time is predominantly determined by two variables: the distance threshold (d) and the quantity of vertices (n) within the trajectory. In contrast, the minimum cluster size, duration, and length parameters are anticipated to exert minimal influence on processing time. The data depicted in Figures 9 and 10 reveal a clear correlation: as the distance threshold expands, the algorithm's processing time lengthens, and as the number of vertices within the trajectory grows, the algorithm's processing time likewise extends.

Figure 8: Images displaying the trajectory of the ball and an associated subtrajectory cluster representing ball movements from the top to the bottom.

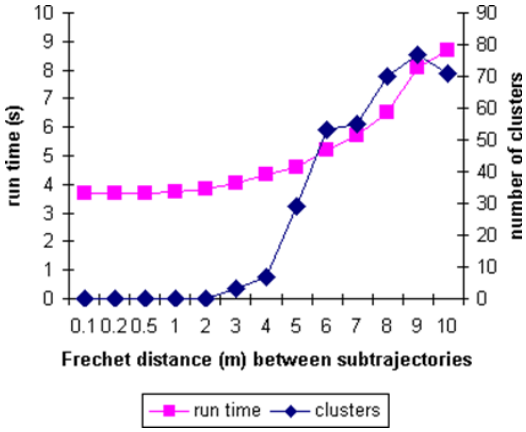

Figure 9: Graph illustrating the relationship between processing time and the quantity of clusters, which varies depending on the selected Frechet distance.

Our experimental findings reveal that the processing time of the subtrajectory clustering algorithm is contingent upon two primary factors: the selected Frechet distance value (d) and the trajectory's scale, quantified by the number of vertices (n). More precisely, when d is heightened, the processing time escalates due to the inclusion of a greater number of vertices in the clustering procedure. Furthermore, an augmentation in the trajectory's size results in an extended processing time, as the algorithm must handle an increased volume of data. The algorithm's theoretical analysis aligns with our experimental results, demonstrating a quadratic relationship with respect to the trajectory's size.

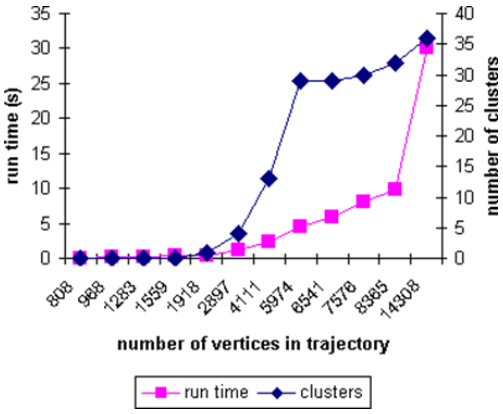

Figure 10: Graph depicting the correlation between processing time and the quantity of clusters relative to the trajectory's size

Our experimental results underscore the substantial influence of the selected Frechet distance on the algorithm's processing time. As this distance increases, a larger number of vertices must be considered, leading to extended processing times.

Furthermore, the trajectory's size has a profound impact on processing time, displaying a quadratic relationship. Additionally, it is evident that a small Frechet distance can hinder the formation of clusters, while a larger distance can yield longer clusters but may impede subsequent cluster formation. Moreover, the utilization of discrete Frechet distance can pose challenges in identifying clusters for smaller trajectory sizes due to the vertex simplification process.

Based on the information illustrated in Figures 9 and 10, it becomes evident that the algorithm holds promise for deployment in an interactive tool when dealing with specific setups and modest inputs. Nevertheless, the existing implementation may not be apt for interactive applications in more intricate scenarios, like those involving extensive inputs or substantial Frechet distances.

It's worth emphasizing that the present implementation prioritizes demonstrating the asymptotic running time behavior rather than optimizing for speed. It serves as a prototype designed for a feasibility study. Employing alternative or more sophisticated algorithms has the potential to yield substantial enhancements in processing speed.

V. CONCLUSIONS

The algorithms and prototypes we have introduced mark the initial strides toward achieving comprehensive automation in football analysis. Though the attainment of complete automation remains uncertain, the provision of additional tools to coaches has the potential to enhance their team's performance.

Our study has demonstrated that employing methods like event detection and clustering can prove fruitful in football data analysis. Nevertheless, there remains scope for enhancements in both accuracy and efficiency. We intend to persist in our exploration and refinement of these tools, with the aim of enhancing their utility for coaches and analysts. Furthermore, our efforts will be directed towards identifying means to streamline the clustering algorithm, making it more efficient.

To fully harness the potential of football analysis and deliver a valuable contribution, it is vital to foster robust and fruitful communication among experts from diverse domains.

REFERENCES

[1] Alt, H., Knauer, C., & Wenk, C. (2004) - This paper discusses an assessment of various distance measures for planar curves, and it's published in the Algorithmica journal (Volume 38, Issue 2, Pages 45–58).

[2] Buchin, K., Buchin, M., Gudmundsson, J., Löffler, M., & Luo, J. (2008) - This work focuses on detecting commuting patterns using sub-trajectory clustering. It was presented at the 19th International Symposium on Algorithms and Computation (ISAAC 2008) and is set to be published in the International Journal on Computational Geometry and Applications.

[3] Fujimura, A., & Sugihara, K. (2005) - This research delves into geometric analysis and quantitative assessment of teamwork in sports, as published in Systems and Computers in Japan (Volume 36, Issue 6, Pages 49–58).

[4] Grunz, A., Memmert, D., & Perl, J. (2009) - This study involves the analysis and simulation of in-game actions through special self-organizing maps. It's published in the International Journal of Computer Science in Sport (Volume 8, Issue 1, Pages 22–36).

[5] Kang, C.-H., Hwang J.-R., & Li, K.-J. (2006) - This research focuses on trajectory analysis for football players and was presented at the Sixth IEEE International Conference on Data Mining - Workshops (ICDMW '06). The proceedings were published by IEEE Computer Society (pp. 377–381).

[6] Lee, J.-G., Han, J., & Whang, K.Y. (2007) - This work explores trajectory clustering through a partition-and-group framework. It was presented at the ACM SIGMOD International Conference on Management of Data (pp. 593–604).

[7] Nanni, M., &Pedreschi, D. (2006) - This study discusses time-focused density-based clustering of moving object trajectories. It's published in the Journal of Intelligent Information Systems (Volume 27, Issue 3, Pages 267–289).

[8] Vlachos, M., Gunopulos, D., &Kollios, G. (2002) - This research focuses on discovering similar multidimensional trajectories and was presented at the 18th International Conference on Data Engineering (ICDE, pp. 673–684).