Web Application Oriented to the Analysis of Multiple Trajectories in Football Matches Based on Georeferenced Data

Martín Cerviño García, Diego Seco Naveiras, and Tirso Varela Rodeiro

Centro de Investigación CITIC, Universidade da Coruña, 15071 A Coruña, Spain
Correspondence: martin.cervino@udc.es

DOI: *https://doi.org/10.17979/spu.23.c46*

Abstract: Data analysis has become an essential tool for understanding and optimizing performance in football. Within this context, the study of player trajectories plays a key role in the tactical analysis of the game. This paper presents the development of a system designed to visualize, search and compare player trajectories during a match based on georeferenced data. Implemented in a web application, the system enables users to identify and explore movements at key moments using either static or dynamic visualizations. In addition, it integrates a system that allows the detection of trajectories with movement patterns similar to a given trajectory, which required the design and optimization of an efficient similarity search algorithm.

1 Introduction

Football performance analysis has undergone a significant transformation in recent years due to the increasing availability of data and advanced tracking technologies. Tactical decisions, player positioning and team strategies can now be analyzed in detail using georeferenced data. Player trajectories, reconstructed from tracking systems, provide valuable insights into both individual and collective behaviors, enabling coaches and analysts to identify key patterns, evaluate strategies and optimize performance. However, access to advanced analytical tools is often limited to professional clubs with substantial resources, leaving lower level teams, researchers and enthusiasts with few options to explore tactical insights.

In this context, the present work develops a web-based system for the visualization, search and comparison of player trajectories in football matches. The system is designed to be intuitive, enabling users to explore key moments, analyze movement patterns and compare trajectories across players. It incorporates a similarity search system to identify trajectories with comparable movement patterns and offers both static and dynamic visualization modes.

The data used for this work were captured by Opta Sportsdata Stats Perform (2025), a company specialized in the collection and analysis of sports data. The information is provided in several structured files, each containing specific aspects of the match. For this study, the dataset corresponds to the match played on February 15, 2023, during the 12th round of the Premier League, between Arsenal and Manchester City. These datasets form the basis for trajectory reconstruction and subsequent analysis.

The main objective of this work is to develop a web application for trajectory visualization and comparison. To achieve this, the system has been designed with several key functionalities: An interactive web interface that allows the visualization of player movements on a football pitch using a map-based viewer. The ability to display multiple trajectories simultaneously, enabling collective analysis such as comparing the movements of defenders against attackers. Filtering based on key match events (e.g., goals, shots, intercepted passes) to restrict the analysis to moments of special tactical relevance. Two visualization modes, one static for fixed trajectory representation and another dynamic, synchronized with match footage. And finally, a similarity-based search system capable of retrieving trajectories with movement patterns comparable to a selected or manually drawn target trajectory.

2 Related work

Currently, some commercial tools offer solutions for football performance and tactical analysis. Catapult (2025) MatchTracker provides 2D visualizations of player positions synchronized with match video. Also, Metrica Sports (2025) offers video-based analysis platforms, enabling users to visualize positions and trajectories, annotate plays and generate clips for detailed study. While these tools provide valuable insights, they do not focus on trajectory visualization. The application presented in this paper distinguishes itself by centering on geospatial trajectory visualization.

3 Developed solution

3.1 Sysytem architecture

The system was developed under a three-layer architecture:

- Client layer: A web interface that provides intuitive interaction with the system. It was built using HTML, CSS, JavaScript and the Leaflet (2025) library for map visualization.

- Server layer: Responsible for managing user requests, processing queries and applying the bussines logic. It was implemented in Python using Flask (2025).

- Data layer: A PostgreSQL database extended with PostGIS (2025) for eficent storage, indexing and querying of geospatial data.

3.2 Data preprocesing and model

The raw dataset consisted of multiple files in different formats (JSONL, JSON and XML), containing tracking data, metadata, event details and match results. A preprocessing stage was required to extract, clean, transform and restructure the information from these files into a consistent representation. To enable efficient querying for visualization and similarity search, a conceptual data model was designed, capturing the main entities of the system: players, teams, frames, events and event types and their relationships (see Figure 1).

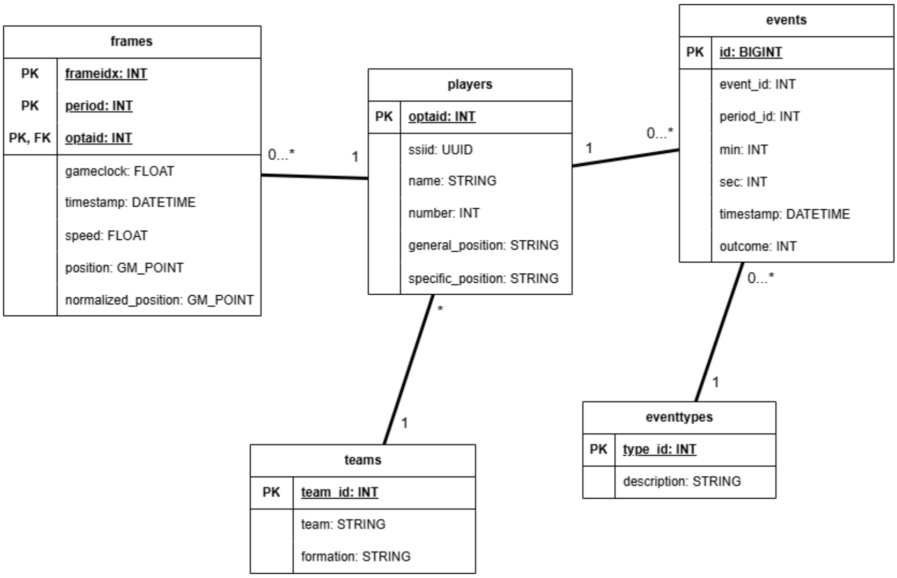

Figure 1: Conceptual Data Model

3.3 Core functionalities

The system integrates some key features. **Trajectory visualization**: users can display individual or multiple player trajectories in static (lines) or dynamic (animated) modes, synchronized with the match video. **Event filtering**: trajectories can be filtered by match events such as goals, shots, or interceptions, enabling analysis in key moments of the match. **Similarity search**: the system allows users to find similar trajectories by searching with either an existing match trajectory or a manually defined one, thanks to the Leaflet Draw extension, which is known as search by sketch. The similarity measure is based on point-to-point Euclidean distance. Given a target trajectory $\tau = p_1, p_2, \ldots, p_n$ and a candidate $\tau' = q_1, q_2, \ldots, q_n$, both of length n, the average Euclidean distance is defined as:

$$d(\tau, \tau') = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{q}_i\| \,, \qquad (37.1)$$

where $\|\mathbf{p}_i - \mathbf{q}_i\|$ *represents the Euclidean distance between the corresponding points* $\mathbf{p}_i$ *and* $\mathbf{q}_i$ *of the compared trajectory, that is:*

$$\|\mathbf{p}_i - \mathbf{q}_i\| = \sqrt{(x_i^{\tau} - x_i^{\tau'})^2 + (y_i^{\tau} - y_i^{\tau'})^2} \qquad (37.2)$$

To optimize the similarity search, the system uses Minimum Bounding Rectangles (MBRs) to reduce search space. Each trajectory is represented by its MBR, defined by the minimum and maximum coordinates along the x and y axes. The target trajectory's MBR is slightly expanded by a margin α to allow a small tolerance and only trajectories inside it are included in the comparison (see Figure 2 for an example of a trajectory's MBR).

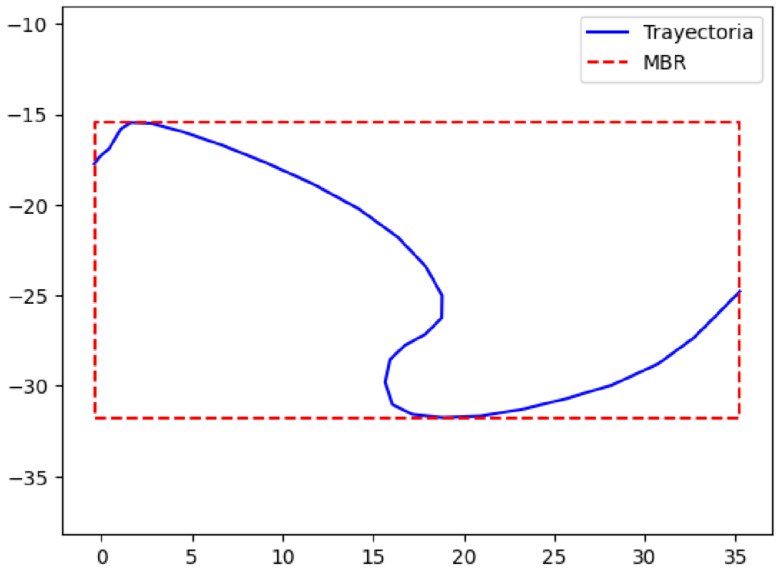

Figure 2: Minimum Bounding Rectangle (MBR)

To compare a target trajectory with those recorded during the match, the system applies a sliding window approach. Each candidate trajectory is divided into subsequences of the same length as the target trajectory. For every subsequence, the average point-to-point Euclidean distance is computed, providing a measure of similarity. After all distances are computed, the subsequences are ranked in order of similarity. Finally, a filtering step is applied to remove subsequences that share temporal intervals, ensuring that the final results are representative and do not simply replicate the same moment. The pseudocode of the implemented similarity search algorithm is presented below in Algorithm 2 .

Algorithm 2 Find k Most Similar Trajectories

1. **Input:** T_{obj}, the target trajectory; DB, the database of trajectories; k, number of results; α, MBR margin
2. **Output:** Set of the k most similar trajectories
3. **Step 1 — Spatial Filtering:**
   a) Compute $MBR_{obj} \leftarrow$ minimum bounding rectangle containing T_{obj}
   b) $CANDIDATES \leftarrow \emptyset$
   c) For each trajectory t in DB:
      i. If trajectory t intersects MBR_{obj} (with margin α):
         A. Add t to $CANDIDATES$
4. **Step 2 — Similarity Search:**
   a) $RESULTS \leftarrow \emptyset$
   b) For each trajectory $cand$ in $CANDIDATES$:
      i. For $i \leftarrow 1$ to $|cand| - |T_{obj}| + 1$:
         A. $sub\_traj \leftarrow cand[i : i + |T_{obj}| - 1]$
         B. $dist \leftarrow$ AverageEuclideanDistance(T_{obj}, sub_traj)
         C. Add $(dist, sub\_traj)$ to $RESULTS$
   c) Sort $RESULTS$ by $dist$ in ascending order
   d) $FILTERED \leftarrow \emptyset$
   e) For each $(dist, traj)$ in $RESULTS$:
      i. If $traj$ does not temporally overlap with any trajectory in $FILTERED$:
         A. Add $(dist, traj)$ to $FILTERED$
5. **Return:** the first k trajectories from $FILTERED$

3.4 User interface

The web application was designed with intuitiveness and usability as central requirements. The main interface consists of two core views:

Trajectory visualization page. The central component of the system. On the left panel, users can filter trajectories by player, position and event type, as well as specify whether they wish to see movements prior to or following the event. The main panel displays the selected trajectories on a football pitch using Leaflet as a map viewer. Static visualization mode presents fixed lines representing player movements, while dynamic mode animates the trajectories synchronized with the match video. Pop-up windows provide contextual information such as player identity, distance covered and average speed.

Figure 3: Trajectory visualization page in static mode

Similarity search page. This view allows users to find similar trajectories to a selected one. The results are displayed in order of similarity and can be visualized and compared.

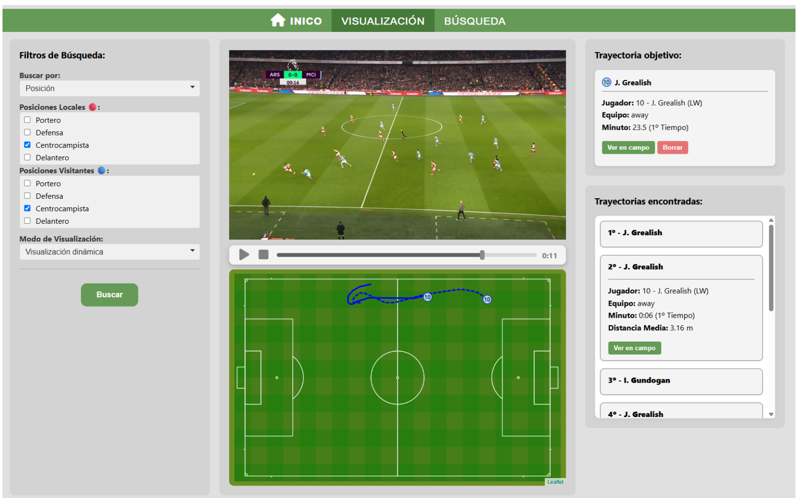

Figure 4: Similarity search page showing the results in dynamic mode

4 Conclusions and future work

This work provides a web application capable of interactively exploring and analyzing player trajectories in football matches. The proposed solution stands out from existing alternatives by combining visualization, analysis and a similar trajectory search system in a single platform. Future work includes extending the system to manage multiple matches, allowing, for example, searches for similar trajectories across an entire season and incorporating additional tactical events, like "high press", "build-up play" and "counterattacks".

Bibliography

Catapult. Catapult — Sports Technology — Unleash Potential. *https://www.catapult.com/*, 2025. [Online; accessed 16-May-2025].

Flask. Flask's Documentation. *https://flask.palletsprojects.com/en/stable/*, 2025. [Online; accessed 18-May-2025].

Leaflet. Leaflet - A JavaScript Library for Interactive Maps. *https://leafletjs.com/*, 2025. [Online; accessed 18-May-2025].

Metrica Sports. Metrica Sports — Cutting-Edge Video Analysis for Coaches & Analysts. *https://www.metrica-sports.com/es/*, 2025. [Online; accessed 16-May-2025].

PostGIS. PostGIS. *https://postgis.net/*, 2025. [Online; accessed 18-May-2025].

Stats Perform. Opta data from Stats Perform. *https://www.statsperform.com/opta/*, 2025. [Online; accessed 16-May-2025].