# Pitch Control Quantification in Soccer: Data Acquisition and Analysis

Balázs Imre
Supervisor: Balázs Csanád Csáji

This report was created as part of the Eötvös Loránd University Applied Mathematics Project Work II course.

***Abstract*—This report presents a project focusing on quantifying and visualizing pitch control in soccer. It begins with the acquisition of match tracking data from the Football Manager 24 (FM24) game, using Python and the Open Computer Vision Library (OpenCV), employing the Hough transform and template matching methods to identify player positions. The analysis then utilizes the Voronoi tessellation, and Javier Fernández and Luke Bornn's pitch control model. The report provides a brief discussion of the implementation of these methods, supported by visualizations. In addition, the final section includes a match analysis to demonstrate the practical application of the models.**

***Index Terms*—Sports data analysis, Soccer analytics, Pitch control, Voronoi tessellation, Player tracking, Data visualization**

## I. Introduction and Motivation

Soccer analytics has traditionally focused on on-ball events, such as pass and shot efficiency or dribble success rate. However, the importance of off-ball events has increased significantly in modern soccer. As Johan Cruyff said, "It is statistically proven that players actually have the ball 3 minutes on average. So, the most important thing is what you do during those 87 minutes when you do not have the ball. That is what determines whether you're a good player or not." [1] This perspective emphasizes the value of analyzing off-ball dynamics. While the methods discussed in this report were developed for soccer analytics, they also have potential applications in other fields, such as traffic management, marketing, and healthcare.

### A. Background

Pitch control refers to the ownership of space by teams. In regions controlled by Team A, the players of that team can act quicker and occupy positions earlier than their opponents. Quantifying the pitch control ratio is crucial for analyzing teams' tactical approaches and evaluating players' abilities to gain an advantage in different areas of the pitch through effective positioning.

### B. Project Goals

We aim to implement two approaches to pitch control: a basic method using Voronoi tessellation, and a more advanced method based on Javier Fernandez and Luke Bornn's concept of player influence area. To achieve this, it is essential to acquire sufficient data, thus we use match tracking data extracted from the Football Manager 24 game (hereafter referred to as FM24).

## II. Data Acquisition and Preprocessing from FM24

One of the main challenges in this project is the limited availability of high-quality, publicly accessible tracking data. For the previous report in Project Work I, the dataset provided by Metrica Sports was sufficient; however, it included only two matches without any additional information (such as teams, player names, etc.). [2] Generating new data is beyond the scope of this project due to the complexity of the game, but the world of video games offers plenty of opportunities to acquire data. The final choice was the FM24 game, since it is widely accepted that this game has the most realistic soccer simulation in terms of player movements and game mechanics. [3] The plan was the following: record the match, identify each player individually, and then locate each player in each given frame.

### A. Recording and Frame Extraction

FM24 provides an opportunity to replay previously played matches in 2D mode, where the soccer pitch is viewed from above, and players are represented by circles containing their shirt numbers. I recorded my screen during these replays and used the resulting video to extract tracking data from the matches using Python and the Open Computer Vision Library. Video settings:

- Format: mp4,
- Resolution: $1920\times1080$,
- FPS: 24.

Game settings:

- Highlight Mode: Full Match,
- Camera: 2D Classic,
- Match Speed During Highlights: Very Fast,
- Match Speed During Text-Only Highlights: Very Fast,
- Match Speed Between Highlights: Very Fast.

### B. Player Detection using Hough Circles

In soccer, each team has 11 players on the pitch, unless a team was penalized with a sending-off. During the game, managers can substitute players; however, once a player was substituted, they are not allowed to return to the pitch. To determine which players participated in the match and to automate the tracking, I captured three screenshots from each match: the first was taken after the start of the first half, the second after the start of the second half, and the third shortly before the end of the game. It is very unlikely that a player who participated in the match would be missing from

all three frames. After that, I applied the Hough transform to identify individual players, and in the later stages, I used these templates to locate players.

The Hough transform is a feature extraction technique used in image analysis and pattern recognition. The goal is to detect noisy instances of lines, circles, and other predefined shapes using a voting procedure. This procedure is carried out in a parameter space, where potential shapes are identified as local maxima. The simplest case is to detect straight lines in a binary image. The line $y = mx + c$ in the image space corresponds to the point $(b, m)$ in the parameter space, and the point $(x_i, y_i)$ in the image space corresponds to the line $c = -mx_i + y_i$ in the parameter space, including all possible parameters of the lines containing the given point. After utilize the parameter space $(m, c)$, we initialize an accumulator array $A(m, c)$, and set $A(m, c) = 0$ for all pairs $(m, c)$. Then, for each $(x_i, y_i)$ point, we increment $A(m, c)$ by 1 if $(m, c)$ satisfies $c = -mx_i + y_i$. (Fig. 1. and Fig. 2.) Finally, the local maxima in the accumulator indicate the most likely line candidates. To handle parallel lines, avoid infinite slopes and large accumulators, trigonometric parameterizations (e.g., using $x \sin\theta - y \cos\theta + \rho = 0$) can be applied.

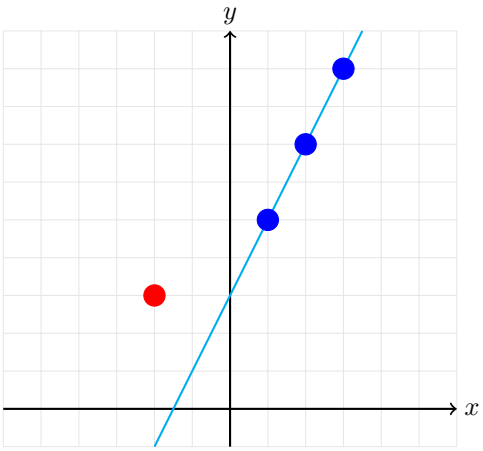


Fig. 1. Hough Transform: Image Space

In our case, each player is represented by a circle. The general equation of a circle is $(x-a)^2 + (y-b)^2 = r^2$ which introduces three parameters: $a$, $b$, and $r$. Thus, the parameter space becomes three-dimensional. Each point $(x_i, y_i)$ in the image space corresponds to a cone in the parameter space defined by $(a - x_i)^2 + (b - y_i)^2 = r^2$. [4]

I used OpenCV's `HoughCircles()` method with the following parameters:

- `method = cv.HOUGH_GRADIENT`,
- `dp = 1`,
- `minDist = 2`,
- `param1 = 50`,
- `param2 = 20`,
- `minRadius = 9`,
- `maxRadius = 11`.

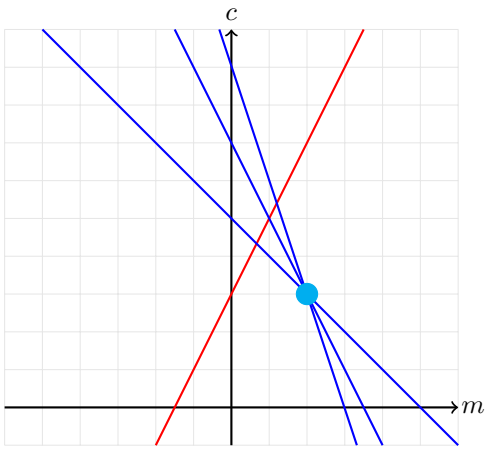


Fig. 2. Hough Transform: Parameter Space

This method uses an advanced version of the Hough transform optimized for detecting circles. After I identified the player circles, I was able to locate the same circles in each frame of the video.


Fig. 3. Identified Player Circles

### C. Template Matching for Player Identification

To find all players, I used the identified player circles as templates. (Fig. 3.) The idea was to iterate through each player and each frame to locate them. Template matching is a method for searching and identifying the location of a template within a given image. For this purpose, I used OpenCV's `matchTemplate()` function. The technique is quite straightforward: it slides the template across the input image – similar to a 2D convolution – and compares the template to each corresponding region of the image.

Let $I$ denote the input image of size $(W, H)$, and $T$ denote the template of size $(w, h)$. The output of the function is a grayscale image $R$ of size $(W - w + 1, H - h + 1)$. Using OpenCV's `minMaxLoc()` function, we can find the position with the highest score, which corresponds to the most likely location of the template in the image.

I tried some metrics, and the best one in terms of reducing false positives was the `TM_CCOEFF_NORMED` metric, defined as follows:

$$R(x,y) = \frac{\sum_{x',y'} T(x',y') \cdot I(x+x', y+y')}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x', y+y')^2}.}$$

In (Fig. 7.) we can see the result of `matchTemplate()` in the case of detecting the player with shirt number 19 from (Fig. 3.).

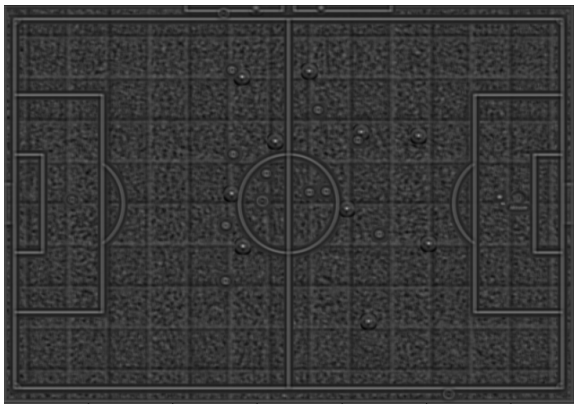

Fig. 4. Match Template Result

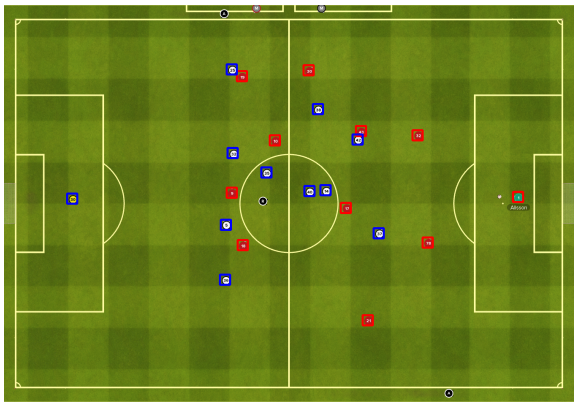

Fig. 5. Detected Players

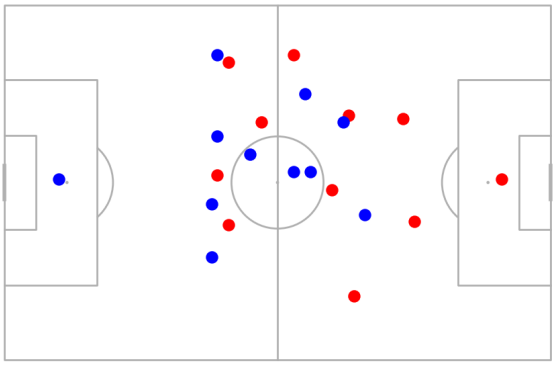

Fig. 6. Extracted Tracking Data

### D. Post-Processing and Dataset Structure

I faced many challenges during the data acquisition process. In this section, we will discuss some of them.

After identifying the player circles using `HoughCircles()` and attempting to locate them with `matchTemplate()`, a significant problem emerged. It was difficult to distinguish players, especially when their circles overlapped. Another challenge was the similarity between certain numbers (e.g., 18 and 19), which appeared nearly identical in some frames. To address this, I attempted to refine the templates by minimizing unnecessary details and emphasizing differences. I experimented with removing the background and cropping the templates, but the most effective method was to apply a circle-shaped mask around the numbers during template matching to highlight their differences.

Despite these improvements, false positives still occurred. To handle this, I introduced a threshold. If the matching score was above $0.95$, I considered it a valid detection. Otherwise, the corresponding location was filled with NaN values. While this solved the misidentification issue, it generated a new problem - how to handle the missing data?

To address this, I used linear interpolation. I filled the missing values by interpolating between the last two known positions of the players. This straightforward method turned out to be quite accurate, as the video had a high frame rate and the gaps with NaN values spanned only a few frames.

In the end, I obtained a dataset with approximately 5 data points per second, tracking player positions with $x$ and $y$ coordinates in the range $[0, 1]$, with the kickoff point located at $(0.5, 0.5)$.

### E. Basic Visualization

Data visualization plays a crucial role in understanding and communicating results effectively. For this purpose, I used Python's *Matplotlib* and *mplsoccer* libraries. Visualization provides insights into strategies and tactics. For instance, plotting the average positions of players allows straightforward categorization into player roles, such as goalkeepers, defenders, midfielders and attackers. (Fig. 7.)

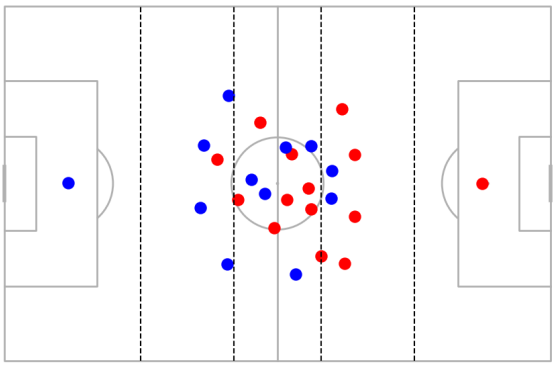

Fig. 7. Player's Average Positions

## III. Pitch Control Quantification with Voronoi Tessallation

The Voronoi tessellation method is our first approach to quantifying pitch control. This technique is intuitive and straightforward. For each player, there is a corresponding region, called Voronoi region, which consists of all points on the pitch closer to that player than to any other.

### A. Formal Definition and Alternative Approaches

The definition and computation of Voronoi tessellation are from Mark de Berg, Marc van Kreveld, Mark Overmars and Otfried Schwarzkopf's Computational Geometry book. [5]

Let $P = \{p_1, p_2, ..., p_n\}$ be a set of points, where each $p_k$ represents a pair of real numbers. The Voronoi region of $p_k \in P$ is defined as:

$$V(p_k) \doteq \{x \in \mathbb{R}^2 \colon d(x, p_k) < d(x, p_l),\ \forall l \in \{1, ..., n\},\ l \neq k\},$$

where $d(x, p_k)$ denotes the Euclidean distance.

In our case, the points $h_1, ..., h_{11}$ and $a_1, ..., a_{11}$ correspond to the players of the home and away teams. Thus, the Voronoi region of player $p_k$ is defined as:

$$V(p_k) \doteq \{x \in F \colon d(x, p_k) < d(x, p_l),\ \forall p_l \in \text{players},\ l \neq k\},$$

where $F$ refers to the set of all points on the field.

We further define $H$ and $A$ as the unions of the Voronoi regions controlled by the home and away players, respectively. These regions can be interpreted as the areas of the pitch dominated by the team.

A more generalized approach for defining influence regions is presented in the work of Tsuyoshi Taki and Jun-ichi Hasegawa. [6] In their method, the Euclidean distance is replaced by the function $t(x, p_k)$, representing the shortest time necessary for player $p_k$ to move from their current position to the point x.

### B. Implementation and Visualization

For the implementation, it was crucial to ensure that the Voronoi regions remained within the boundaries of the pitch. To address this, the players' positions were reflected across the four boundaries of the pitch. This ensured that the bisectors of the original and reflected points lay on the pitch edges, thereby constraining the Voronoi regions to the field.

In the following visualization (Fig. 8), we observe a scenario in which the away team is building an attack from the back, while the home team is defending. The red points represent the home team's players, the blue points denote the away team's players, and the yellow point indicates the position of the ball. One of the home team's strikers is pressing the away team's defender who has possession. The away team is attempting to create passing options in wide areas, while the home team maintains a compact defensive shape in the center of the pitch. This reflects a zonal defending strategy — instead of marking players man-to-man, the defending team focuses on protecting important zones. The Voronoi regions support this observation, since the central areas of the pitch are dominated by the home team (red zones), while the wide areas are controlled by the away team (blue zones).

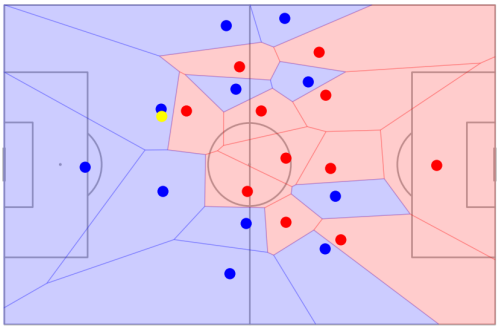

Fig. 8. Voronoi Regions Visualization

## IV. Pitch Control Quantification with Player Influence Areas by Fernandez and Bornn

Javier Fernandez and Luke Bornn introduced an advanced method to measure space generation and occupation in their article, enabling to evaluate player's off-ball movements. This relies on the concept of player influence areas. [1]

### A. Model Description

A player's influence on nearby areas depends on several factors, such as their location, velocity, and distance to the ball. Players farther from the ball have influence over larger areas, as they have more time to reach the ball within a wider region. Furthermore, player's velocity is also a crucial factor, players sprinting at full speed in a particular direction exert greater influence on the corresponding area compare to those walking or jogging.

The influence of player $k$ at a given location $x$ and time $t$ is defined as

$$I_k(x, t) \doteq \frac{f_k(x, t)}{f_k(x_k(t), t)},$$

where $x_k(t)$ refers to the position of player $k$ at time $t$, and $f_k(x, t)$ is the density function of a bivariate normal distribution. The covariance matrix and expected value dynamically change based on the player's velocity, direction, and distace from the ball. The exact mathematical formulation can be found in the appendix of their article. [1]

### B. Implementation and Visualization

I implemented their approach by creating a mesh grid representing points on the pitch, and aggregated the influence values for each player at each grid point.

We now revisit the previous scenario. (Fig. 9). The green arrows represent the players' directions and velocities, which provide additional context compared to the Voronoi diagram. Due to visualization constraints, these arrows are illustrative and do not reflect exact velocity values. The resulted plot highlights the controlled areas; while home team controlled areas are represented by higher values, away team controlled areas are shown with lower values. This dynamic view of the game reveals a slight shift in player movements toward the less crowded side of the pitch. The wide players of the

away team are moving into open space to receive the ball and help the attack, whereas the home team remains compact, with players staying close together in the central areas to force the opposition to play backwards or into less dangerous wide zones.

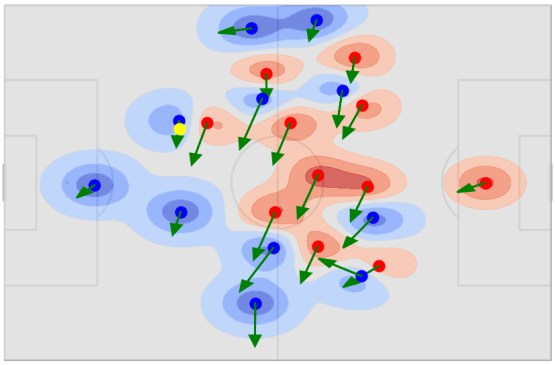

Fig. 9. Player's Influence Regions

## V. Match Analysis and Future Work

### A. Match Analysis

In this section, we present a brief analysis of a simulated UEFA Europa League group-stage match where Liverpool (Premier League) hosted Linzer Athletik-Sport-Klub (Austrian Bundesliga). Liverpool had already secured qualification for the knockout stage, whereas LASK needed a win to keep their hopes of qualifying. Given these circumstances, Liverpool had a slightly rotated squad, resulting a relatively balanced contest.

Using the Voronoi tessellation approach, it was straightforward to quantify the area controlled by each team. For the player influence area model, I introduced a threshold to define controlled regions: a region was considered controlled by the home team if the aggregated influence exceeded the threshold, and by the away team if it fell below the negative threshold.

In the following figure, we can see the average pitch control ratio over 5-minute intervals, highlighting territorial dominance throughout the match. (Fig. 10.) The $x$-axis denotes time in minutes, while the $y$-axis represents the pitch control ratio. Vertical lines indicate the timing of goals — red for the home team and blue for the away team. The match concluded with a 3–2 victory for Liverpool, with goals in the 9th, 50th, and 81st minutes for the home team, and in the 35th and 76th minutes for the away team.

The chart reveals a correlation between pitch control and scoring opportunities. Liverpool exerted pressure early in the first half, which resulted an early goal. To response, LASK increased their attacking intensity, concluding in improved pitch control and an equalizing goal in the 35th minute. The second half began with dominance from the away team; however, Liverpool regained the lead via a counterattack in the 50th minute. This goal disrupted the away team's rhythm, and the subsequent 5-minute interval favored the home side.

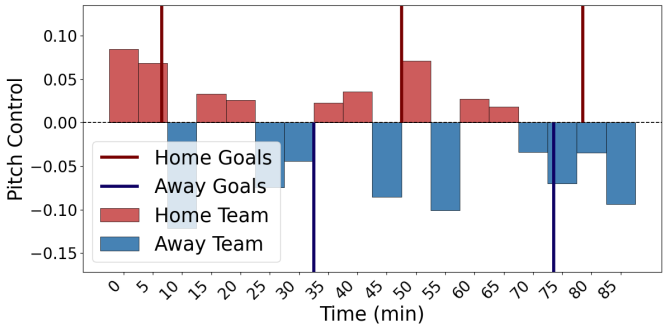


Fig. 10. Pitch Control Ratio - Match Analysis

Later, LASK pushed forward again and managed to equalize in the 76th minute. Despite being under pressure at that stage, Liverpool reclaimed the lead in the 81st minute.

The next figure provides a spatial breakdown of average pitch control values across the pitch. (Fig. 11) Lower values indicate areas controlled by the away team, while higher values correspond to zones under home team control.

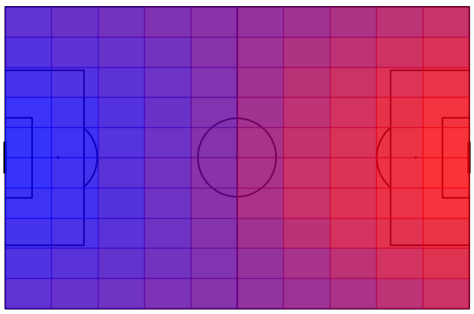

Fig. 11. Average Pitch Control - Match Analysis

The midfield zones display relatively balanced values in the 0.4–0.6 range, slightly favoring Liverpool. This suggests that they maintained a compact defending shape and were effective in disrupting the away team's build-up play. This central dominance aligns with their zonal defending strategy discussed earlier, where they prioritized controlling key central areas over man-marking opponents. We can also observe that the away team had more control in the wide areas.

### B. Next Steps

Since we are able to generate data using FM24, we can perform further analytics by leveraging more data.

One direction could be to quantify the value of different points on the pitch. For example, space occupied near the opponent's goal is far more valuable than space occupied by a team's goalkeeper. Developing metrics to evaluate the value of different regions could be beneficial for advancing soccer analytics methods.

## REFERENCES


[1] Javier Fernandez and Luke Bornn. Wide open spaces: A statisticial technique for measuring space creation in professional soccer. *MIT Sloan Sports Analytics Conference*, 2018.

[2] Metrica Sports. Metrica sports sample data. https://github.com/metrica-sports/sample-data. [Online; accessed 04-Dec-2024].

[3] Sports Interactive. Football manager 24 [game].

[4] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[5] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer, 2000.

[6] Tsuyoshi Taki and Jun-ichi Hasegawa. Quantitative measurement of teamwork in ball games using dominant region. *International Archives of Photogrammetry and Remote Sensing*, 5:125–131, 2000.