Football analysis using spatio-temporal tools

Joachim Gudmundsson*
University of Sydney and NICTA
Sydney, Australia.
joachim.gudmundsson@sydney.edu.au

Thomas Wolle
Arclight Investments Pty. Ltd.
Sydney, Australia
thomas.wolle@gmail.com

ABSTRACT

Analysing a football match is without doubt an important task for coaches, talent scouts, players and even media; and with current technologies more and more match data is collected. Several companies offer the ability to track the position of the players and the ball with high accuracy and high resolution. They also offer software that include basic analysis tools, for example straight-forward statistics about distance run and number of passes. It is, however, a non-trivial task to perform more advanced analysis. We present a collection of tools that we developed specifically for analysing the performance of football players and teams.

Categories and Subject Descriptors

F.2.0 [**Analysis of algorithms and problem complexity**]: General

General Terms

Algorithms

Keywords

football analysis, spatio temporal analysis

1. INTRODUCTION

Ten years ago the first regular occurrences of automated analysis of football games appeared [2, 11]. Computer vision and video image processing had by that time reached a level at which football players and the ball could be tracked by video with enough accuracy for the games to be further analysed by coaches and media. There are several companies that provide sports analyses services, mainly focussing on football, for example *Amisco* [2] and *Tracab* [11].

Most companies in the industry provide similar services. High-definition cameras are mounted around the pitch and during the game all objects on the pitch are tracked 10-25 times per second and the coordinates are delivered in real-time (with some variations). During a post-processing step the data is annotated with additional information either manually or semi-automatically. The extra information added is typically: free kicks, throw-ins, corners, offside, kick-offs and red/yellow cards. In some cases even passes and tackles are added manually. This is the data generation part, which is also the part that traditionally has received the most attention and research resources. Amisco and Tracab also developed various analytical software provided to coaches, talent scouts, media and referees to perform analysis on the generated data. This part has generally been neglected and in comparison with the image processing part it is technologically behind in the development. The existing software packages provides very simple statistical information about the match and the performance of the players. Typical information provided is distance covered by players, top/average speed of players, number of passes performed by players, number of shots on goal, free kick ball speed, heat maps of players, player average position, offside margin and so on. This is all easily obtained immediately from the data without any sophisticated algorithmic tools or advanced processing of the data.

It is apparent that there is a strong demand for more advanced analytical tools. The top European football teams and the national teams deal with the lack of analysis tools by employing game analysts to record the games and then perform the required analysis manually.

The situation is similar in academia, that is, there is a vast amount of literature on tracking objects from video and hardly any research on analysing the obtained movement data. However, scattered results are available and in recent years more research is done in the area. See for example Grunz et al. [6] and Kim et al. [8].

1.1 Related results

We believe our algorithms and implementations belong to the still immature research area that aims at automated football analysis. This is a recent area of research but it is becoming more popular as can be seen by the increasing amount of related work from various fields of computer science. For example, Kang, Hwang and Li [7] proposed a method to quantitatively evaluate the performance of football players. Their approach is based on four different measures that include different regions for each player and the kicks the players perform. Grunz, Memmert and Perl address the analysis of actions and detecting (movement) pat-

*Funded by ARC grant FT100100755.

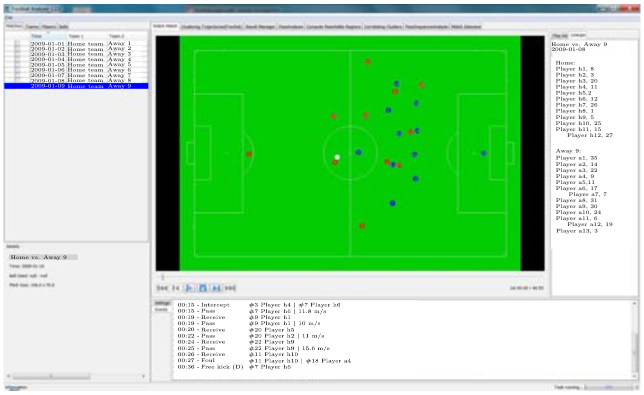

Figure 1: The graphical interface of the FA tool.

terns [6] in a football match using self-organising maps and neural networks. Duch et al. [5] aimed to rate the quality of a team by using the following principle. The more alternatives a team has for a ball to travel and finish on a shot, the better that team is. And, the more times the ball goes through a given player to finish in a shot, the better that player performed. This can be modelled using networks analysis, see for example the "Football Interaction and Process" model and the software system developed by Beetz et al. [3]. Recently Kim, Kwon and Li [8] presented a framework where they define a feature model to specify the basic units of analysis; this gives a set of morphological properties of each feature type for which a set of operations are defined.

Relatively much research has been done on defining and computing the region a player is "dominating" on the field. Taki and Hagesawa [10] initialized the fundamental concept of dominant region which is defined as "a region where the player can arrive at earlier than others". At a particular given time, the dominant regions of the players are computed based on their current positions, speeds and constant accelerating abilities. There are later studies that further developed more realistic models of calculating the dominant regions, see Nakanishi et al. [9].

Our results.

In this paper we present a set of tools that we developed for analysing football players and teams. In the rest of the paper we will list the tools, present their functionalities and briefly discuss the aim of each tool. Note that due to the page limit this paper focuses on the problem statements and the functionality of the tools. A picture of the GUI is shown in Fig. 1. On the left side a user can select the data to be used in the analysis, such as matches, teams or players. The selected data is then used in the area on the right. This area contains several tabs; roughly speaking one tab per analysis tool.

2. PASSING ANALYSIS

Among the most basic parts of a football game and the performance of a player is the passing. We provide two tools for passing analysis; the first tool calculates all possible passing alternatives (Section 2.1) at each point in time and the second tool computes frequent pass sequences (Section 2.2).

We only implemented tools for passes in 2D but we believe the basic ideas and algorithms can be extended to 3D.

2.1 Passing analysis

The first tool we will discuss can be used to evaluate a player's passing abilities; executing a pass, receiving a pass and also the ability to "see" an opening for a pass. The latter is due to the fact that we do not only have information about all passes a player makes, but also information about every time the player did not make a pass.

A *pass* is defined by its start coordinates, the direction of the pass and the initial speed of the pass. Informally we say that a pass is *valid* if its performed by a player p and can be reached by a player in the same team as p before any other player. This concept will be discussed further below.

The tool provided in the FA software works as follows:

Input: A set of 23 trajectories representing the movement of the ball and 22 players on the pitch during one or several football matches.
Output: For every point in time when some player has control of the ball the tool outputs a description of every possible valid pass.

The basic idea of the tool is simple. Given the position of the ball and the players at time t_0, calculate all possible passes and give a rating to the hardness of every pass. The tool consists of two main parts; the motion model used to define a *passable* area and the underlying algorithm that computes the passable areas.

2.1.1 Usefulness

Given an algorithm for the above problem, it can be used for several interesting problems.
• *Rate how hard a pass is to execute:* Rating depends on the speed interval and the directional angle for which the pass would be successful, together with the speed and direction of the passer. (The less precise a pass has to be in terms of speed and direction to be valid, the easier it is to perform.)
• *Rate a player's ability to decide which pass to make i.e., the players overview of the game:* For each pass made by the player the chosen pass is compared with the available passing alternatives at that time. The exact rating is then a trade-off between the difficulty of the pass and the possible advantage gained from a successful pass.
• *Rate the player's ability to execute a pass:* Considers every passing attempt made by the player together with the rating of that pass.
• *Rate a player's ability to move into a position where he is open for a pass:* Considers a combination of the total time the player is open for a pass, the rating of a possible pass and the possible advantage gained from a successful pass.
• *Rate a player's ability to receive a pass:* For each attempt of a pass to the player the rating depends on the change of direction and speed that is required by that player to intercept the pass before anyone else and stay in control of the ball after receiving it.

2.1.2 Motion model and passable area

Intuitively the passable area for a player is a geometric region around a player in which that player could receive a pass. Such a region would depend on the motion model and the position of the ball and all the players. Closely related is the concept of *dominant regions* defined by Taki and Hasegawa [10]. They defined the dominant region of a player to be the area on the field that the player can reach before any other player (similar idea to a weighted Voronoi diagram). The dominant region depends on the motion model.

The GUI contains a tab where the motion model can be chosen. In the current version only three basic motion models have been implemented. However, any new model can easily be incorporated into the system.

The first two models are very similar to the motion models in [10], while the third model is based on historical data of a player. We will focus our discussion on this model since it is new. The general idea is to use the historical movement data from a player to generate a personalized motion model. That is, given a time period, say t seconds, one can construct a convex polygon that contains all the points that the player can reach within t seconds given his direction and speed at time t_0. This region is called the player's *reachable* region. Growing t from t_0 defines a motion model for that player. Note that each sample in the trajectory of a player can be seen as a starting time. We are ready to define the passable regions, which is similar to the dominant regions defined by Taki and Hasegawa [10].

DEFINITION 1. *A player p is* open for a pass *if there is some direction and (reasonable) speed the ball can be passed in such that p can intercept the ball the quickest.*

Intuitively this makes sense; if a player can reach the ball before anyone else when it is passed with a certain speed and direction then no other player could possibly get the ball in this circumstance.

Given the positions, speeds and direction of motion of the soccer players on the field, our tool computes who is *open*, and for what angles and speeds of the ball.

2.2 Pass sequence analysis

The second passing tool detects sequences of passes occurring frequently. For example, it could be that a common way to transport the ball from defending 1/3 of the field to the attacking 1/3 of the field is often done using the same players. Or that 2-3 players know each other well and therefore pass to each other more often than to other players. This tool would detect these kind of passing patterns.

Input: A team T and an ordered sequence S containing all passes made by players in T during one or several football matches.
Output: A data structure D such that given a query (τ, o) the data structure returns all permutations of τ players in T such that the ball is passed from a player p_1 to a player p_τ via players $p_2 \dots p_{\tau-1}$ at least o times.

2.2.1 Usefulness and Algorithm

Given an algorithm for the above problem it can be used for several problems, for example:
• *Analysis of how attacks are built up:* Considers all attacks and reports the most common player combinations and clusters formed by the ball movement from defence to an attacking position.
• *Passing combinations patterns between regions:* Detects the most common player combinations involved in passing the ball from region A to region B.

Consider a team T and let S be the ordered sequence of all passes made by players in T during one or several football matches. The sequence S can be seen as a set of strings each describing the sequence of passes between the players in T. A string ends when the opposing team gains control of the ball, or the game is interrupted by an event (free-kick, throw-in, etc.). For example, if player p_1 passes player p_3 who passes player p_2 who looses control of the ball to the opposing team then the string would read $p_1p_3p_2$.

The algorithm finds all frequent patterns of length at least τ that occurs at least o times. The underlying data structure is a standard suffix tree, that is, a data structure that presents the suffixes of a given string in a way that allows many problems on strings to be solved quickly. The suffix tree for a string S is a tree whose edges are labeled with strings, such that each suffix of S corresponds to exactly one path from the tree's root to a leaf. Constructing such a tree for the string S takes time and space linear in the length of S for a constant size alphabet, which in our case is 11. After the suffix tree has been constructed it is very easy to process a query. Consider a query (τ, o) that should return all permutations of τ players in T such that the ball is passed from a player p_1 to a player p_τ via players $p_2 \dots p_{\tau-1}$ at least o times. This corresponds to all paths in the suffix tree ending at level τ that has frequency o, which can be reported in time linear with respect to the number of reported permutations.

3. CLUSTERING AND CORRELATIONS

An important part of the game is the players' movement on the pitch. Next we present a tool that clusters players movement, and then, in Section 3.2, clusters will be analysed to discover movement correlations between players.

3.1 Clustering movement

Given a trajectory or a set of trajectories we look for subtrajectory clusters, which are movements of a player, or players, that are repeated often. To simplify the description we will assume that the input is one single trajectory. The formal definition comes from [4].

DEFINITION 2. *Given a trajectory T with n vertices, a subtrajectory cluster, denoted $C_T(m, d, \ell)$, for T consists of at least m non-identical subtrajectories $T_1, \dots, T_m$ of T such that the time intervals for two subtrajectories overlap in at most one point, the distance between the subtrajectories is at most $2d$, and at least one subtrajectory has length ℓ.*

The tool provided in the FA software works as follows:

Input: A player's trajectory T from one or several games, and three parameters m, d and ℓ.
Output: Every subtrajectory cluster C, such that C contains at least m subtrajectories, the distance between any two subtrajectories in C is at most d and the length of the longest subtrajectory in C is at least ℓ.

3.1.1 Usefulness and Algorithm

Given an algorithm for the above problem it can be used, for example, for the following questions:
• Reports the most common movement patterns formed by the ball transported between defense and offense.
• Report the most common movement patterns of a single player, or several players.

We present a prototype where a user can specify certain parameters of the cluster, and then the prototype will reliably detect the clusters according to the chosen parameters. One way to evaluate the usefulness of the clustering is by examining the results visually, i.e. drawing the clusters together with the trajectory into an image of a football pitch.

In the above definition we never defined how the 'distance' between two curves is measured. It can be defined in many different ways and in this paper we decided to use the discrete Fréchet distance.

The Fréchet distance can be intuitively explained in the following way: Imagine a person walking their dog on a leash. The person will follow a certain trajectory or path T_p, while the dog follows a different path T_d. The discrete Fréchet distance between T_p and T_d is the smallest length of a leash that allows the person and the dog to walk on their paths, where the person and the dog can change their speed or even pause, but they are not allowed to backtrack. The discrete Fréchet distance considers only positions of the leash where its endpoints are located at vertices of the two polygonal curves and never in the interior of an edge.

We implemented the algorithm in [4] that groups parts of trajectories into clusters such that the discrete Fréchet distance between any two members of a cluster is small. The running time of the algorithm is roughly quadratic for most realistic scenarios.

3.2 Correlating clusters

The aim of this tool is to detect correlations between the most frequent movements of the players. We do this by calculating correlations between subtrajectory clusters computed from the movement (trajectories) of the players.

The tool provided in the FA software works as follows:

Input: A set T of subtrajectory clusters.
Output: All correlations in time between the subtrajectory clusters in T.

The basic idea of this tool is very simple. Assume we are given two subtrajectory clusters, for two different players A and B. We can detect correlations in the movements of A and B along the clusters if there are several subtrajectories for A and B that overlap in time, that is, their movements along the clusters occurred at the same time (at least partly).

3.2.1 Usefulness and Algorithm

An algorithm for the above problem would typically answer questions of the following type:

- Detect correlations between midfielders and attackers when building up the game.
- Detect how units in the team move together, for example the defensive line, or the left winger and the left defender.
- Detect how the team move together when loosing or gaining possession.

We present an algorithm where a user can specify a set of subtrajectory clusters together with two parameters specifying the number of players and the number of correlations. Correlation can be defined in several ways and our implementation only considers one.

We define two concepts of cluster correlation namely: local correlation and global correlation. When only considering the temporal aspect, a subtrajectory can be seen as a time interval whose start time and end time are the time of the subtrajectory's start point and end point, respectively. Hence, from now on we can view a subtrajectory cluster as a set of time intervals. Subsequently, the event of two players' movements occurring simultaneously can be interpreted as two time intervals (belonging to different players) overlapping. Hence, we say that $k > 1$ time intervals are *locally correlated* if they overlap (at least at one point) along the time-axis.

Based on the definition of local correlation, we define the notion of *global correlation* which considers the number of occurrences of local correlation sets among clusters. We argue that k players are correlated if they repeat their simultaneous movements frequently. Thus, if the number of local correlation sets (λ) between k clusters is larger than or equal to a defined threshold number (θ), we say that they are globally correlated.

Acknowledgement

We thank all the students that have been involved in this project: David Rizzuto, Cahya Ong, John Jiang, Suzy Howlett, Huynh Chi Dai, Merlin Zhang and Sidong Liu.

4. REFERENCES

[1] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. International Journal of Computational Geometry and Applications, 5(1-2):75-91, 1995.

[2] Amisco. www.sport-universal.com

[3] M. Beetz, B. Kirchlechner and M. Lames. Computerized Real-Time Analysis of Football Games. IEEE Pervasive Computing, 4(3):1536-1268, 2005.

[4] K. Buchin, M. Buchin, J. Gudmundsson, M. Löffler and J. Luo. Detecting Commuting Patterns by Clustering Subtrajectories. International Journal on Computational Geometry and Applications, 21(3):253-282, 2011.

[5] J. Duch J. S. Waitzman and L. A. N. Amaral. Quantifying the Performance of Individual Players in a Team Activity. PLoS ONE, 5(6):e10937, 2010. Public Library of Science.

[6] A. Grunz, D. Memmert and J. Perl. Tactical pattern recognition in soccer games by means of special self-organizing maps. Human Movement Science, 31(2):334–343, 2012.

[7] C.-H. Kang, J.-R. Hwang and K.-J. Li. Trajectory analysis for soccer player. IEEE Spatial and Spatio-temporal Data Mining (SSTDM), 2006.

[8] H.-C. Kim, O. Kwon and K.-J. Li, Spatial and Spatiotemporal Analysis of Soccer. Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2011.

[9] R. Nakanishi, J. Maeno, K. Murakami and T. Naruse. An Approximate Computation of the Dominant Region Diagram for the Real-Time Analysis of Group Behaviors. RoboCup 2009, pp. 228-239, 2009.

[10] T. Taki and J. Hasegawa. Visualization of dominant region in team games and its application to teamwork analysis. Proc. of Computer Graphics International, pp. 227-235, 2000.

[11] Tracab. www.tracab.com