Soft Comput
DOI 10.1007/s00500-011-0733-0

ORIGINAL PAPER

Performance analysis in soccer: a Cartesian coordinates based approach using RoboCup data

Pedro Henriques Abreu · José Moura · Daniel Castro Silva · Luís Paulo Reis · Júlio Garganta

© Springer-Verlag 2011

Abstract In soccer, like in business, results are often the best indicator of a team's performance in a certain competition but insufficient to a coach to asses his team performance. As a consequence, measurement tools play an important role in this particular field. In this research work, a performance tool for soccer, based only in Cartesian coordinates is presented. Capable of calculating final game statistics, suisber of shots, the calculus methodology analyzes the game in a sequential manner, starting with the identification of the *kick* event (the basis for detecting all events), which is related with a positive variation in the ball's velocity vector. The achieved results were quite satisfactory, mainly due to the number of successfully detected events in the validation process (based on manual annotation). For the majority of the statistics, these values are above 92% and only in the case of shots do these values drop to numbers between 74 and 85%. In the future, this methodology could be improved, especially regarding the shot statistics, integrated with a real-time localization system, or expanded for other collective sports games, such as hockey or basketball.

Keywords Soccer heuristics definition · Game events detection · Cartesian coordinates system · Player and ball position

1 Introduction

Soccer is one of the collective sports games (CSG) with more participants and supporters all over the world (it is played by over 240 million players in 1.4 million teams and 300 thousand clubs around the world) (Acar et al. 2008). In this particular sport, 2 teams, with 11 players each, try to reach the objective of scoring at least one more goal than the opposing team, thus achieving the victory in the match. According to Grehaigne et al. (1997), the essence of the game can be described as: a team must coordinate its actions to recapture, conserve and move the ball so as to bring it within the scoring zone and to score a goal. Training a team is mainly a task of enhancing team performance by providing feedback about the performance of the athletes and team (Hughes and Bartlett 2002). Human observation and memory are not reliable enough to provide accurate and objective information from athletes in high-performance competitions.

In most team sports, an observer is unable to view and assimilate the entire action taking place on the playing area, due to its attention to the game critical areas, which makes most of the peripheral play action to be lost (Hughes et al. 2001). During a soccer match, the coach can become the recipient of a great amount of information; as a result,

P. H. Abreu (✉) · J. Moura · D. C. Silva · L. P. Reis
Laboratory of Artificial Intelligence and Computer Science, Department of Informatics Engineering, Faculty of Engineering of Porto University, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
e-mail: pha@fe.up.pt

J. Moura
e-mail: ei08172@fe.up.pt

D. C. Silva
e-mail: dcs@fe.up.pt

L. P. Reis
e-mail: lpreis@fe.up.pt

J. Garganta
Investigation Center of Investigation, Education, Innovation and Intervention in Sport, Faculty of Sport of Porto University, Rua Dr. Plácido Costa, 91, 4200-450 Porto, Portugal
e-mail: jgargant@fade.up.pt

Published online: 06 May 2011

he might not be able to evaluate and objectively exploit all the technical and tactical elements that may come along (Frank and Miller 1991). Emotional factors, such as stress and anger, or even more subjective aspects such as prejudice, can also lead to a decrease in concentration, and consequential misinterpretation of the game reality (Carling et al. 2007). In consequence of that, many are the coaches that tried to collect all this information through automatic performance analysis systems. Two different types of analysis can be identified in a soccer match—notational and motion analysis. Notational analysis is based on event collection, either during the game of using a post-game analysis process, for classification and performance evaluation (Franks 1996). Motion analysis is focused on raw features of an individual's activity and movement during a soccer match, without attempting any qualitative evaluation (Carling et al. 2007). Analysis can focus on four main categories—behavioral (mental factors that can be assessed from body language and other action), physical (movement and biometric data can be measured to improve training), tactical (includes choosing the appropriate strategy and tactics to be used against a specific opponent) and technical (skills such as passing can be assessed to improve training and performance) (Carling et al. 2007).

Recent technology developments promoted a proliferation of research projects that tried to give an effective support in this particular area (soccer coach analysis). Camera support systems (CSS) is one of the most well known technologies researchers have worked with over the past years, despite of the computational demands and occlusion problems, which are still two big issues for this kind of approach. Other research works defined only static concepts of soccer in order to classify some video sequences (see Sect. 2 for more detailed information about these topics).

Another research project that has emerged is the RoboCup competition,[1] constituted by several leagues and whose main goal resides in developing a team of autonomous humanoid robots that are capable of defeating a human soccer team (this research project is explained with more details in the next section). In spite of this project having reached important results in its various leagues, in the area of game analysis, most research works have based their calculation method of game statistics in the set of algorithms used by the RoboCup Soccer Simulation Server. These algorithms are spread over thousands of lines of code and no formal representation of them exists. Also, it is very difficult to make a simple change in a specific concept within the code.

In this research project, the authors tried to solve the soccer experts requirements (previously mentioned). For that, a new set of flexible soccer heuristics is proposed in order to calculate final game statistics related to the technical category of match analysis (using automatic notational analysis). Based only on players' and ball's Cartesian coordinates in a soccer game, the use of these heuristics enables the detection of a range of individual and collective events regarding each team, representing their strengths and weaknesses. These events were defined by an expert group of soccer researchers, which makes this study closer to the soccer day-to-day reality. Using a sequential analysis of a soccer match, the calculation method is based on the identification of an increase of the ball velocity vector. This event is designated as a *kick*, which is the basis for all heuristics. Also, this set of heuristics allows the user to make some transformations concerning its definitions, such as altering soccer field regions, ball dimension, restrict a match action to a specific area, changing event definitions, and others.

To validate our approach, RoboCup Soccer logs of the 2007 and 2009 2D Simulation League competitions were used, due to the lack of real (human) soccer data. However, all the logic used in the calculation algorithms is independent of the nature of the data being used.

The remainder of this paper is organized as follows: Sect. 2 describes the related work in the soccer area, especially regarding to soccer ontologies and languages. Section 3 presents the system architecture and all concepts involved in the proposed set of soccer heuristics. Section 4 exposes the achieved results and finally, in the last section, conclusions are presented and future work trends are discussed.

2 Related work

Nowadays, a soccer coach's work is evaluated mainly by the number of wins that his team achieves in a certain competition. In consequence of that, professional soccer coaches try to use the best measurement tools available to improve their athletes' performance in the next match. In recent years, many are the research projects that emerged to improve the gathering of information about the performance of the players and the team in a match or competition. In the next subsections, a set of investigation projects aiming to help soccer agents to interpret, in the best possible manner, the game actions to be analyzed, are divided in three distinct areas—sports video analysis, Sports ontologies and the RoboCup competition.

2.1 Sports video analysis

Over the past years, in the sports video area, researchers have focused their work in problems like: rule-based

[1] More information available online at http://www.robocup.org/.

semantic classification (Tovinkere and Qian 2001), scene reconstruction (Yow et al. 1995), shot detection and classification (Ekin and Tekalp 2003), highlights extraction and event detection (Xu et al. 2003) and structure analysis (Xie et al. 2001, 2004). These approaches use a recording of the match for a post-match, off-line processing and analysis.

Nowadays, two different standards exist for audiovisual applications and content descriptions—MPEG-7 (Smith 2001) and TV-anytime (Sklansky 1982). Based on these, Tsinaraki et al. (2003) proposed a framework in order to manage the semantic metadata used for audiovisual information, providing a semi automatic creation of MPEG-7 metadata description as well as TV-anytime descriptions. Based also on MPEG-7, Yu et al. (2006) proposed a hierarchical organizing model to represent high- and low-level information in sports video.

2.2 Sports ontologies

The concept of ontology appeared in the 80's, but only in the 90's did it become popular among researchers (Gruber 1995). For Guarino (1998), an ontology is a set of entities and relationships among them, that capture knowledge on a specific application domain. Also, for this author, the domains where ontologies can be useful include knowledge engineering, knowledge representation, information retrieval and extraction, qualitative modeling, knowledge management and organization, database design and language engineering. However, this type of approach had not yet included the world wide web. In consequence of that, McGuinness (1998), enhanced online search using ontologies and, in 1999, the Dublin Core initiative (Dublin Core Metadata Initiative 1999) defined a set of metadata elements for cataloging library items. Nowadays, ontologies are present not only in academic environments, but also in the business world, where they represent an important role in many online applications, such as e-commerce (Amazon), search (Yahoo) and others (McGuinness 2003).

Over the past years, in the soccer area, two distinct ontologies have emerged, proposed by Ranwez and Moller, respectively. Ranwez's research work is focused on construct narratives abstraction, based on a set of related events (Crampes et al. 1998). In 2002, Ranwez et al. defined a soccer ontology (Ranwez Soccer Ontology[2]) that represents a set of soccer concepts like rules, actions or player attributes (name, nationality, and so on). This language is used to support video annotations in a soccer match. In 2004, a new soccer ontology is proposed by Moller (SWAN Soccer Ontology[3]) to be integrated in the SWAN (Semantic Web Annotator) project, which consists in automated extraction of metadata from natural language web content, and presents many additional concepts representing agents within the game as well as a number of concepts surrounding a soccer match. This project also uses the KIM platform (Popov et al. 2003), which is based on GATE[4] (Cunningham et al. 2002).

As a summary for this subsection, it is relevant to note that ontologies are capable of representing some static soccer concepts but cannot represent dynamic situations in a specific time frame, like for instance determine which team will execute a throw in. In this work the ontologies concepts were used only to formally represent the concepts from the soccer experts panel.

2.3 RoboCup competition

RoboCup is a scientific and educational international joint project (Kitano 1998a, b) to promote research in Artificial Intelligence, Robotics and related fields. The main research areas present in this competition are related to autonomous agents, multi-agent systems, real-time reasoning and sensor fusion, among others, to enable robots to play soccer games.

Since the first official RoboCup competition in 1997, in which over 40 teams participated (real and simulation combined), there have been many other international and regional competitions.

Nowadays there are many different leagues divided into five main classes: RoboCup Soccer, RoboCup Rescue, RoboCup@Home, RoboCupJunior and Demonstrations. The Soccer league competitions started in 1997 with a very clear goal: to develop a team of fully autonomous humanoid robots that could defeat the human soccer world champion by 2050. Integrated in this class, the RoboCup 2D Simulation League uses a simulator (RoboCup Soccer Simulator Server) based on a client-server architecture (Chen et al. 2003), and each agent (team players) connects to the server in order to play; The game may be viewed through a monitor (RoboCup Soccer Simulator Monitor) that represents the simulation in a graphical manner. The server generates a log file for each game, which can be viewed in a log playing tool (RoboCup Soccer Simulator Log Player).

Coach competition was one of the challenges present in RoboCup Soccer until 2006. The main idea behind this competition was to develop a Coach Agent that could perform an intelligent high-level analysis of the game. In consequence of that, the need to have a communication

[2] Ranwez, S. Ranwez Soccer Ontology. DAML Ontologies Library Web site, submitted in 2002, available online at http://www.daml.org/ontologies/273.

[3] Moller, K. SWAN Soccer Ontology, submitted in 2004, available online at http://sw.deri.org/~knud/swan/ontologies/soccer.

[4] More information available online at http://gate.ac.uk/.

system that could use a clear semantic between the coach agent and the players emerged.

In 2001, Reis and Lau (2002) created a language called COACH UNILANG, which enables high-level communication between a coach agent and a team. This language uses some high-level concepts like field regions, time periods, tactics, formations, situations and player types (especially important in competitions that use heterogeneous agents).

In 2007, the RoboCup federation adopted their official language—CLANG (Chen et al. 2003) which, specifies low-level concepts and tries to combine them with high-level soccer concepts, and is used in the competitions for communication between coach and players.

One example of an application developed using the coach agent is proposed by Gonzalez et al. (2008), which consists in using a soccer agent to calculate final game statistics. In spite of the fact that some of these applications present a large set of statistics (sometimes, however, not following the definition presented by the soccer experts panel) they base their calculations in the soccerserver flags, which prevents them from expanding their applications to other environments (such as other collective sports) and also from adding further statistics to the tool.

In the past years, other research areas have appeared in the 2D Simulation League. One of these areas is the automatic description of a soccer match. Using a commentator system, it is possible to generate real-time reports for arbitrary matches of the RoboCup simulation league. In this particular point, three approaches will be analyzed—Byrne, multi-agent interactions knowledgeably explained (MIKE) and Rocco (Andr et al. 2000). All three approaches use as input data the same information the RoboCup Soccer Simulator Monitor receives for updating its visualization, such as:

- Player—location (Cartesian coordinates within the field), orientation (body and head orientation), energy parameters (stamina, effort, recovery) and viewing capabilities (width and quality), among others, of all players;
- Ball—location (Cartesian coordinates);
- Game—current simulation cycle, game state (throw-in, offside, corner, goal kick and others), team names and current score.

Looking more minutely at each approach, the Byrne system generates appropriate affective speech and facial expressions, based on game analysis data and on the character's personality, emotional state, commentator's nationality or the team it supports, among other types of information, and uses a face model as an additional means of communication (Binsted et al. 1998). The MIKE system (Tanaka-Ishii et al. 1998) is a real-time commentator system that supports three distinct languages: English, Japanese and French. The main capability of this system is to identify interactions between players in order to classify team behaviors and to generate predictions concerning the short-term evolution of a given situation.

The Rocco (RoboCup Commentator) is a continuation of a research project that appeared in the 1980's called Soccer (Andre et al. 1988), which, using natural language, tried to interpret a scene in a restricted domain. Rocco is a TV-style live commentator which uses the RoboCup Soccer Simulator Monitor combined with an emotional spoken description of a specific scene.

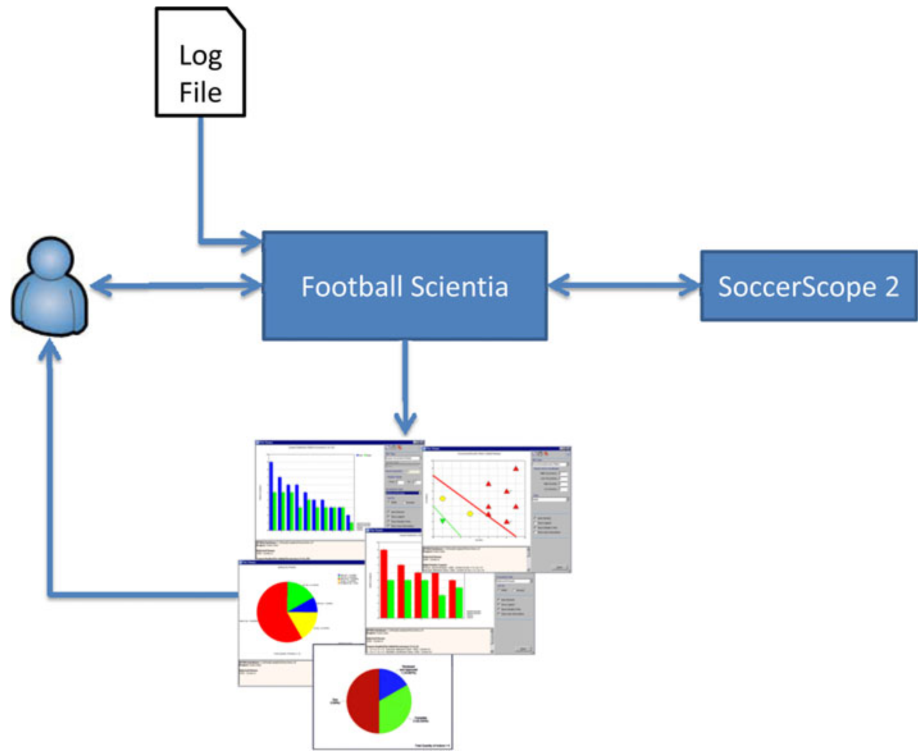
A comparison between these three systems is illustrated in Table 1.

In conclusion, although in recent years many research works have emerged that provide some soccer static definitions (like the commentator projects analyzed above), or in some cases capable of identifying some soccer events, currently, a formal set of heuristics with a set of dynamic and static soccer terms capable of automatically detecting and calculating game events does not exist. Also, some works have been developed in order to model agent behavior based on observation (see Rozinat et al. 2009; Ledezma et al. 2004; or Stone and Veloso 2000). However, none of those approaches based their knowledge in high level statistics, instead using for their studies only information related to scored goals. This research work tries to eliminate the gap between those type of team performance

Table 1 Comparison Between Commentator Systems (based on Andr et al. 2000)

Commentor system	Analysis	Natural language generation	Output
Byrne	Observers' recognize events and states	Templates, marked up for expression and interruption in real-time	Expressive speech and facial animation
MIKE	Events and states (bigrams, voronoi)	Templates, interruption, abbreviation using importance scores	Expressive speech
Rocco	Recognition automata	Parameterized template selection + real-time nominal-phrase generation	Expressive speech

Fig. 1 System architecture



improvement studies by presenting an approach capable of calculating high level statistics than can constitute a remarkable tool to soccer coaches, in order to improve their teams' performance. Also, it is important to state that the knowledge presented by the developed tool is exclusively based on ball and players' coordinates, thus eliminating possible errors provided by the RoboCup official tool and presenting excellent results. Finally, the set of calculated statistics was defined by a soccer experts board, which constitutes another important improvement in this type of approaches.

3 Approach

In this section, the project's architecture is explained, some technological and implementation considerations are made, and the soccer heuristics concepts are introduced and detailed.

3.1 System architecture

This research project was implemented using the architecture depicted in the diagram present in Fig. 1. The user starts his interaction with the Football Scientia module by selecting the log file he wants to be processed, which is then passed along to the already existing SoccerScope 2 software. This module is responsible only for loading the log file, and returning the information contained in that file in a structured form. The Football Scientia module, using this information and a sequential analysis process (explained in the following subsection), produces a series of statistics according the set of defined heuristics (also detailed below).

3.2 Implementation details

In order to load the robotic log files, the SoccerScope 2[5] software is used. Written in Java, this software presents a well structured code with an extensive and comprehensive documentation, that provides new users with a rapid learning curve. However, it supports only version 3 of the 2D Simulation League log files (2007 competition or earlier). In order to use the 2009 log files in this research work, which are created using version 4, a tool present in the RoboCup Soccer Simulator Log Player package[6] is used to convert them into version 3. Also, as the project was developed using Ruby language (due mainly to the experience of the authors with such language), it was necessary to find a bridge between our implementation and the SoccerScope's reusable components. To establish this connection, JRuby[7] was chosen. This language is the implementation of Ruby in Java

[5] More information available online at http://ne.cs.uec.ac.jp/~koji/SoccerScope2/index.htm.

[6] More information available online at http://sourceforge.net/apps/mediawiki/sserver/index.php?title=Download.

[7] More information available online at http://jruby.codehaus.org/language.

and proved to be a mature language, representing an important role in this work.

The majority of soccer events, with the exception of those related to game breaks, such as faults or forced breaks to provide assistance to an injured player, have similarities. At the origin of this kind of events is always an increase of the ball velocity or a change in the direction of ball's motions (named a *kick*), which can represent various events, like a pass, shot, and so on. Equation 1 shows this concept, where t_1 and t_0 are instants of time and V_{ball}, D_{ball} are ball velocity and direction, respectively.

$$
\begin{aligned} kick(t_0) \leftarrow & \, ||\mathrm{V}_{\text{ball}}(t_0)|| < ||\mathrm{V}_{\text{ball}}(t_1)|| \wedge t_1 = t_0 + 1 \\ & \vee D_{\text{ball}}(t_0) \neq D_{\text{ball}}(t_1) \end{aligned} \tag{1}
$$

The soccer game was organized in an array (Fig. 2), each position representing a cycle in the game (in the robot soccer, a game consists of 6,000 cycles, so in this particular situation the array will have 6,000 positions), and containing information about the players and ball position, player energy and vision, among others. For this research work, only the players and ball positions are used in the detection of game events.

Before the game analysis can be performed, the user has to specify which are the events he wants to be detected. After that, the sequential array analysis starts. In the first loop through the array, all *kicks* are detected and marked, representing the possible events that occurred in the game.

In the second step, the positions following any given *kick* (and before the next detected *kick*) are analyzed according to second and third rules (constraint and final condition, respectively (see Abreu et al. 2010), which normally leads to a reduction in the number of events (many *kicks* correspond to dribble events, which are not detected by this application) (Algorithm 1). Also, a class for each event type was created, as to allow for a more flexible and modular internal programming architecture, which also enables the application to take advantage of parallel processing capabilities.

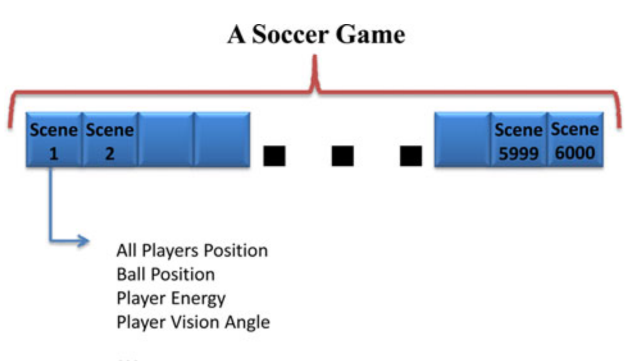


Fig. 2 Soccer game structure

Algorithm 1 Generic Event Detection Algorithm

```
for Cycle i = 0 to max_Cycles − 1 do
  if (kick.start_condition (scene [i] , scene [i + 1])) then
    addKick (i)
  end if
end for
for all Event Class event do
  for all Cycle i in Kicks do
    for j = i + 1 to nextKick do
      if (! event.constrain (scene [j])) then
        break
      end if
    end for
    if (event.final_condition (scene[j])) then
      addEvent (event, i)
    end if
  end for
end for
```

3.3 Soccer field regions

With the purpose of increasing the information quality of the statistics calculation and to better define some soccer concepts, several regions were defined as illustrated in Fig. 3.

For each midfield (defensive and attacking), a set of 16 regions were defined. All these regions are described using relative coordinates and suffer a 180° rotation for the second half of the game. Also, some global variables are defined, like ball size (relevant in goal detection situations) or margins of the field (important to detect events that occur when the ball exits the playable area).

In order to transform the process of specifying the soccer field division into a more flexible one, five dynamic variables (X_n, X_f, Y_x, Y_n, Y_f) were created, as illustrated in Fig. 3, allowing the user to quickly change the previously establish divisions without loosing features in the soccer heuristics. X_f and Y_f represent the field dimensions, X_n the penalty box length, Y_n the penalty box wing width and Y_x the wing area width. Other areas, like the middle areas (back, center and front) or the wings (back, middle and front), are obtained using these variables as a basis.

3.4 Soccer heuristics

In this section, all terms included in the proposed heuristic soccer event detection algorithms are explained and justified.

3.4.1 Pass information

Generically, in a soccer game, there are two types of passes—successfully executed ones, which means that a pass is well performed between two teammates, and missed passes (sometimes called wrong passes), which results in a ball recovery by the opponent team.

Having the *kick* as a basis, the detection of a successful pass consists in identifying two consecutive *kicks* performed by teammates, and detecting the moment (between

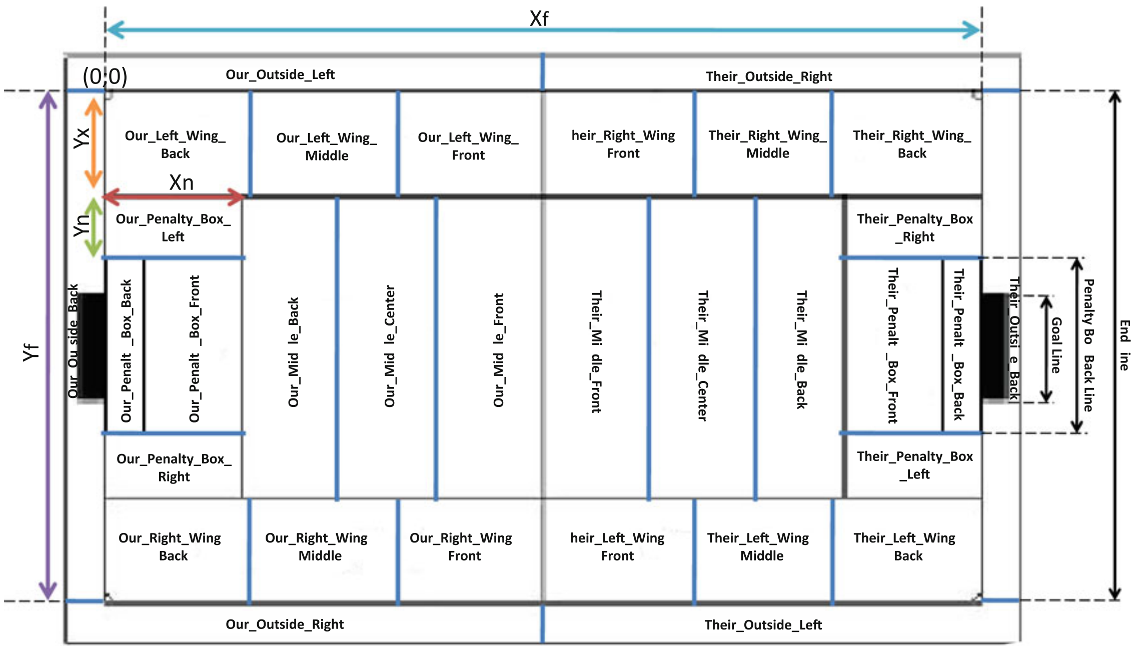


Fig. 3 Soccer field regions divisions

the two *kicks*) when the second player receives the ball, as depicted in Eq. 2, where P_0 and P_1 are the passer and receiver players, respectively, P is a generic player, t_0, t_1 and t_2 time instants, and b is the ball. Also *dist* (b, P) indicates the distance between the ball a P player.

$$\begin{gathered} SuccessfulPass(P_0,P_1,t_0,t_1) \leftarrow KicksBall(P_0,t_0) \\ \wedge KicksBall(P_1,t_2) \wedge SameTeam(P_0,P_1) \wedge t_1 > t_0 \\ \wedge t_2 > t_1 \wedge P_0 \neq P_1 \\ \wedge (\neg\exists(P,t) : t > t_0 \wedge t < t_2 \wedge KicksBall(P,t)) \wedge \\ (\neg\exists(P,t) : t > t_1 \wedge t < t_2 \wedge P \neq P_1 \wedge dist(b,P) < dist(b,P_1)) \end{gathered} \tag{2}$$

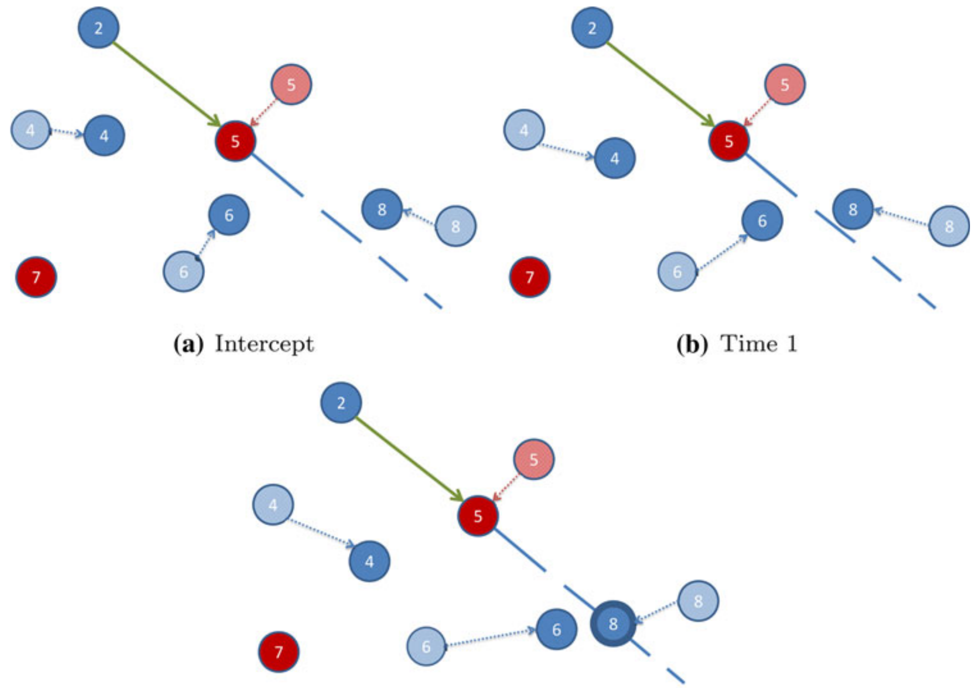
The main issue in this type of events is to detect when a player receives the ball. The initially adopted solution was based on a proximity algorithm, that after detecting the first *kick*, analyzes the area within a circle, centered on the ball (which reduces the number of possible players to test for possession), and detects the player closest to the ball (which will be considered the ball receiver). In spite of this solution being used in many research works (and often considered as a classical one), some issues are still present. The main problem consists, in some occasions, in the detection of the correct receiver when other players are near the ball's trajectory. This situation is the result of a wrong correlation between ball possession and ball proximity concepts, which have different meanings. In consequence of that, the initially used algorithm was adapted into a new approach, based on the detection of two consecutive *kicks* (Algorithm 2).

After receiving or intercepting the ball, the player, after some cycles, will execute a *kick* (which can be a pass, a shot, or even dribble). It's assumed that if a player does not touch the ball, he never had it under his control. So, to detect the existence of a pass, the analysis was centered in detecting a *kick* from the possible receiver of the ball. After detecting the second *kick*, the only issue remaining is to find when he gained possession of the ball, which can be calculated by performing a reverse temporal analysis and applying the proximity algorithm discussed above. If this player is the same that performed the first *kick*, the event is a dribble (not included in this analysis); if the player belongs to the same team as the one who performed the first *kick*, a successful pass is detected; otherwise (the player does not belong to the same team), a missed pass is detected, which can represent several distinct scenarios, explained below.

Algorithm 2 Pass Detection Algorithm

for all Cycle i in $Kicks$ **do**
  $cycle \leftarrow false$
  $originalKicker \leftarrow getPlayerKicking(i)$
  $secondKicker \leftarrow getPlayerKicking(nextKick(i))$
  if $originalKicker \neq secondKicker \wedge sameTeam(originalKicker, secondKicker)$ **then**
    for $j = nextKick - 1$ **to** $i + 1$ **do**
      for all Player p in Players **do**
        if $p \neq secondKicker \wedge distance(p, Ball, j) < distance(secondKicker, Ball, j)$ **then**
          $receiveCycle = j + 1$
          $cycle \leftarrow true$
          $break$
        end if
      end for
      if $cycle$ **then**
        $break$
      end if
    end for
    $AddSuccessfulPass(originalKicker, secondKicker, i, receiveCycle)$
  end if
end for

Fig. 4 Example of distance algorithm



(a) Intercept **(b)** Time 1

(c) Time 2

In the case of a missed pass being detected, in most situations (except for a shot in the direction of the goal line), it is important to determine which was the player that would most likely have received the ball. To that purposes, an algorithm was developed based on ball motion detection and prediction. The path traveled by the ball (between the first *kick* and the moment when either a second player receives the ball or it leaves the play field) is used to determine its direction and velocity, which are then used to simulate its path (from the moment the first *kick* was executed and beyond the receiver player or up to the play field outline, respectively). With this information, the distance between the ball and every player (of the same team as the one that performed the *kick*) is calculated for each cycle of the path (Algorithm 3), and a probability of a player reaching the ball is associated. This probability is inversely proportional to the distance between player and ball, and includes a multiplying factor that decreases with the temporal distance from the initial *kick* (given that long passes are usually less likely to occur, especially in robot soccer). The players' positions considered in this algorithm are determined considering an ideal motion toward the ball, in an interception course. If only one player could actually intercept the ball, it is chosen as the most probable receiver for the ball; otherwise, the player with the highest probability of intercepting the ball is chosen.

Algorithm 3 Distance Calculation Algorithm

$IniCycle \leftarrow$ Cycle of Initial Kick
for all Cycle c in $BallPath$ **do**
  for all Player p in $Teammates$ **do**
    $dist \leftarrow distance(p, Ball, c, IniCycle)$
    $addToProbabilityVector(p, Probability(dist, c - IniCycle))$
  end for
end for
return $HighestProbability(ProbabilityVector).Player$

A more graphical example of this algorithm is represented in Fig. 4. In this particular scenario, the blue player (number two) tries to execute a pass for a teammate. However, the ball was recovered by a red player (number 5) (Fig. 4a). In consequence of that, according the ball movement, the algorithm traces a virtual path that would have been executed by the ball if it had not been intercepted by the red player. After that, and as easily seen in Fig. 4b and c, for each possible ball position, in each cycle, the distances between the ball and each player are calculated according not only to the previous player position, but also to an estimate for a new player position in a specific cycle.

The situation presented in Fig. 4 is very interesting because if the algorithm was only based on the distance between the players and the ball, the teammate that is closest to the ball is player number six (in the first instance after his team lost the ball—Fig. 4b). However, analyzing the ball movement, and doing an estimation for all player movement, the conclusion is that the player that had a higher probability to receive that pass was player number 8.

3.4.2 Shot information

A shot occurs when a player shoots the ball, in his attacking field, in the general direction of the opponent's goal line, and with enough initial velocity that allows the ball to reach it. Equation 3 shows these conditions, where $\mathbf{A}_{\text{ball}}$ represents the acceleration of the ball, $Vel(b, t_0)$, $Pos(b, t_0)$ represents the ball velocity and position in an instance t_0, respectively, and, P represents a player.

$$\begin{aligned} &Shot\,(P, t_0) \leftarrow Belongs\,(P,\ team) \wedge KicksBall\,(P, t_0) \\ &\wedge InRegion\,(Pos\,(P, t_0), AttackField\,(team)) \wedge \\ &\big(\exists(t) : Pos\,(b, t_0)_X + Vel\,(b, t_0)_X t \\ &\quad + \frac{{\mathrm{A}_{\mathrm{ball}_X}}^2}{t} 2 > X_f \ \wedge Pos\,(b, t)_Y > 0 \ \wedge Pos\,(b, t)_Y < Y_f\big) \end{aligned} \tag{3}$$

Three distinct events are detected that can be considered a shot: a Shot on Target, an Intercepted Shot and a Shot. A Shot on Target occurs when a player kicks the ball in the goal direction and the kick has enough strength to make the ball reach the goal line adding tolerance (0.5 m) around the goal for each side. On the other hand, if the ball does not have the goal direction as defined previously (after kicked by a player) but still leaves the field through the Penalty Box area (and is not in conditions to be considered a shot on target), this event will be marked as a shot. Finally if an opponent player intercepts the ball after the first player kicked the ball (with all the conditions to be classified as a shot target or a shot), this event will be classified as an intercepted shot.

3.4.3 Goal scoring

To win a soccer match, one team needs to score at least one more goal than its opponent. A goal occurs when the ball, in its totality, passes through the goal line. Implementation constraints, however, make this condition insufficient by itself—the ball, after being kicked by a player, can occupy the same position it would after a goal, but crossing through the penalty box back line instead. So, the detection of the goal must also analyze if the ball actually intercepted the goal line before completely leaving the play field, using the path of the ball for that purpose (see Algorithm 4).

It is important to note that in this project, when a goal is detected, no other event is considered (namely the TargetShot and the Outside events).

Algorithm 4 demonstrates the differences between a Goal, a Target Shot and an End Line Shot.

3.4.4 Outside information

When a ball moves outside the play field, three situations can occur: throw-in, corner or goal kick. The only factor that distinguishes these situations is the region where the ball left the play field and the last player that kicked the ball. In order to detect which team will have ball possession, the algorithm analyzes which player executed the last *kick* before the ball left the play field—Algorithm 5 (the *calculateOutsideType*() function determines the type of outside: throw-in, corner or goal kick).

Algorithm 4 Goal, Target Shot and End Line Shot Detection Algorithm

```
for all Cycle i in Kicks do
  kicker ← getPlayerKicking(i)
  for j = i + 1 to nextKick − 1 do
    A = BallPosition(i);
    B = BallPosition(j);
    if InRegion(BallPosition(j + 1), OutsideBack) then
      if Intercepts(A, B, GoalLine) then
        AddGoal(kicker, i)
      else if Intercepts(A, B, GoalLine + 0.5) then
        AddShotonTarget(kicker, i)
      else if Intercepts(A, B, PenaltyBoxBackLine) then
        AddShot(kicker, i)
      end if
    end if
  end for
end for
```

Algorithm 5 Outside Detection Algorithm

```
for all Cycle i in Kicks do
  kicker ← getPlayerKicking(i)
  for j = i + 1 to nextKick − 1 do
    if InRegion(BallPosition(j), Outside) then
      outsideType = calculateOutsideType
      (BallPosition(j))
      AddOutside(kicker, i, outsideType)
    end if
  end for
end for
```

3.4.5 Offside information

The offside rule was established in 1924 and is probably the most difficult real-time situation to detect in a soccer match. In this work, an offside is defined in a similar way to a pass event. However, to detect this event, two conditions must be verified (Algorithm 6): a successful pass must be detected and the player that received the ball must have been in an invalid field position at the moment of the pass. A position is considered invalid to receive a pass if the receiver, at the moment of the pass, is the last player before the goalie, and the ball's motion has a component in the direction of the end line. Also, a player is only considered to be in an invalid position if he is in his attacking midfield.

During the game, if an offside situation occurs but the referee does not mark it (because an opponent player intercepts the pass), this type of event is classified as an Intercepted Offside (position validation rules are applied to the receiving player) (see Algorithm 6).

Algorithm 6 Offside Detection Algorithm

```
for all Cycle i in Kicks do
  originalKicker ← getPlayerKicking(i)
  secondKicker ← getPlayerKicking(nextKick(i))
  if !(sameTeam(originalKicker, secondKicker)) then
    receiver = DetermineReceiver()          \\Algorithm3
    if InvalidPosition(receiver, i) then
      AddInterceptedOffside(receiver, i)
    end if
  end if
  if originalKicker ≠ secondKicker ∧ sameTeam(originalKicker, secondKicker) then
    if InvalidPosition(secondKicker, i) then
      AddOffside(secondKicker, i)
    end if
  end if
end for
```

4 Results

In order to validate the soccer concepts described in the previous section, a set of twelve 2D Simulation League games (6 from 2007 and 6 from the 2009 competition) were selected. The criteria for choosing these particular games were the competition phase where they occurred and the variety of results that were observed: high goal differences (4 games), small goal differences (<3 goals—5 games), and draws, with and without goals (1 and 2 games, respectively). Also, the set of selected final game statistics was divided in four distinct groups: **Pass, Shot, Offside, Goal and Outside**. Validation was performed having a manual classification process as a basis—this classification was done by a board of soccer experts.

Starting with a global analysis of the events selected for detection during the analyzed matches, Table 2 shows that successful passes are the most common event in the games, followed by missed passes (with less than half of the occurrences of the former). All other events present a much lower frequency, with outsides standing out as the most frequent of these events (approximately 16 outsides per match).

Regarding pass results (both successful and missed passes), as can be seen in Table 3, more than 98% of the total successful passes and more than 96% of missed ones were correctly detected (with a standard deviation of 2.09 and 4.39% and presenting an average of 3,33 and 3,08 of passes observed but not detected per game named Difference in Table 3, respectively). It is important to note that validated missed passes include not only a correctly detected missed pass but also the most likely destination player. Thus, these results constitute good indicators, especially regarding two complex factors in calculation, such as the detection of ball possession and the calculus of the destination player in the missed pass.

Table 2 Average frequency of events

	Successful passes	Missed passes	Shot	Intercepted shot	Shot on target	Goal	Outside	Intercepted offside	Offside
High goal difference (4)	234	105.75	7.5	12	5.00	10.75	13.75	2.25	2.75
Small goal difference (5)	227	100.2	3.4	3.4	1.00	1.8	15.20	3.00	4.20
Draw (3)	155.67	75.67	3.0	4.33	3.33	0.67	23.00	0.67	2.67
Average (12 games)	211.5	95.92	4.67	6.5	2.92	4.5	16.67	2.17	3.33
Standard deviation	39.88	25.07	5.48	8.61	5.55	6.36	7.09	2.76	3.47

Table 3 Accuracy results for successful and missed passes

	Successful passes				Missed passes			
	Percentage detected	Average of sucessful passes observed	Difference	False positive	Percentage detected	Average of missed passes observed	Difference	False positive
High goal difference (4)	99.25	234	1.75	0.25	96.93	105.75	3.25	0.25
Small goal difference (5)	98.15	227	4.2	0.2	98.00	100.2	2	0.2
Draw (3)	97.43	155.67	4	0.33	93.83	75.67	4.67	0.33
Average (12 games)	98.34	211.5	3.33	0.25	96.60	95.92	3.08	0.25
Standard deviation	2.09	39.88	3.98	0.45	4.39	25.07	3.53	0.45

Regarding the different groups of games, one can see that matches with a higher number of goals also have a higher number of passes, both successful and missed, with those matches that ended in a draw presenting a considerable difference in comparison to the other games. Also, the average number of total successful passes observed is approximately 2.1 times higher than missed ones.

In what concerns shots, Table 4 shows the results for Shots, Intercepted and Shot on Target, where one can see that the average detection percentages are between 74 and 85%.

These results can be explained by two main reasons, intrinsically linked to the robot soccer reality. The first reason, that explains the low number of shots that occur during matches, is related to robot soccer teams strategies, that invariably attempt to score a goal by a combination of passes until the player is almost sure of reaching the objective of scoring. The second reason, that justifies the software's lower detection rate, is related to a somewhat rare situation that can occur during the match—two players can be at equal distance from the ball, when a *kick* is detected, both in a position to kick the ball, as depicted in Fig. 5 (both players 2 and 7, of opposing teams, could have kicked the ball, resulting in completely different events being detected). This makes it extremely difficult for a Cartesian coordinates based system to accurately detect all *kicks*. Given that shots are far less common than passes, these situations assume a higher percentage of occurrences, thus contributing to the presented results.

Table 5 presents the statistics related to goal and outside detection. All 54 goals scored in all 12 analyzed games were successfully detected by the software.

In relation to the outsides, more than 93% of all situations were also confirmed. As mentioned, outsides include three situations: throw-ins (when the ball leaves the field through one of the side lines), corner kicks and goal kicks (when the ball leaves the field through one of the end lines). The goal kick situation includes all End Line Shots and part of Target Shots (the other part being when an opponent player catches the ball in the Penalty Box Back Area).

It is important to note that the few situations the software was unable to automatically detect were all related to specific game situations that lead the RoboCup Soccer Simulation Server to mark an outside even before the ball crosses the field line. If the server detects that the ball's trajectory is leading it outside the play field, and that no player can reach it in time, an outside is marked, and the ball either moves to the destination place (corner, goal kick place, or the side line) or stops before leaving the field. The outside is announced through a server status variable, and the game continues, thus making it difficult for a Cartesian coordinate based system to correctly identify the event.

Table 4 Accuracy results for shot, intercepted and shot on target

	Shot				Intercepted shot				Shot on target			
	Percentage detected	Average of shots observed	Difference	False Positive	Percentage detected	Average of intercepted shots observed	Difference	False positive	Percentage detected	Average of shots on target observed	Difference	False positive
High goal difference (4)	70	7.5	2.25	0.75	70.83	12	3.5	0.5	85	5	0.75	0.25
Small goal difference (5)	76.47	3.4	0.8	0.4	82.35	3.4	0.6	0.4	80	1	0.2	0
Draw (3)	77.78	3	0.67	0.33	76.92	4.33	1	0.66	90	3.33	0.33	0
Average (12 games)	74.64	4.67	1.25	0.5	77.16	6.5	1.67	0.5	84.17	2.92	0.42	0.083
Standard deviation	21.69	5.48	0.97	0.522	26.60	8.61	2.67	0.522	24.69	5.55	0.90	0.288

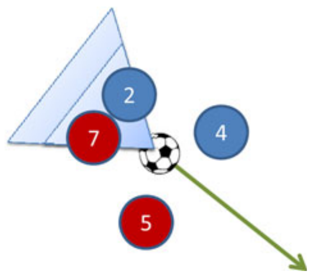


Fig. 5 *Kick* player detection

Further analyzing the results, one can see that games with more even teams (draws and games with a small goal difference) have a higher number of outside situations. A higher number of outsides also leads to a lower percentage of useful play time, since a number of game cycles (approximately 100 cycles) are allowed for each of these situations, and thus, games with a higher number of outsides usually have a lower number of other types of events, such as passes (the most common event in the game).

Regarding offsides, more than 97% of the intercepted offsides and almost 93% of offsides were well detected, which constitutes good results in this particular scenario. All four situations where the software failed to correctly identify an offside are related to the situation depicted in Fig. 5 and already described above Table 6.

Figure 6 presents a unified view on all statistics, and for all three types of games. It is relevant to note that only in three situations were the results of <92%: the three types of shots.

These results show that, even using only Cartesian coordinates as a basis for the detection of events, our approach proved to be an efficient one to solve this specific problem.

Doing a comparison between this tool and other tools presented in Sect. 2.3, it is important to state that before this research work there was no tool capable of automatically detecting a large set of soccer statistics using only Cartesian coordinates. Also, none of the existent tools was able to calculate high level statistics—an example of that is the shot statistic (another example could be the offside statistic); in this research work, three types of shot are defined: shot, intercepted shot and shot on target, as explained in Sect. 3, and this type of division constitutes a novelty in these approaches.

5 Conclusions and future work

In this section, conclusions that can be drawn from this work are discussed and future work trends are presented.

In this research project, a set of formal soccer heuristics is presented. As referred in Sect. 3, several soccer events are defined based on the detection of an event called *Kick*, which is related to the increase of the ball velocity. The results achieved in the validation process, and discussed in the previous section (in two-thirds of the cases attaining over 92% of detection accuracy), show that the use of a sequential analysis process in the detection of soccer events constitutes a good approach to this kind of problems. Unlike other researches' works that base the identification of events in the game status, given by the RoboCup Soccer Simulation Server, thus limiting the events that can be detected, our approach is capable of automatically identifying a larger set of game events (as is the case of an intercepted offside). Also, this method presents additional information, such as the identification of the player that would receive an intercepted pass (with an associated degree of certainty) and also allows the user to change, in a restricted form, some definitions in the heuristics, related to ball dimension, field regions and the rules for identifying events (pre- and post-conditions and constraints). On a more technical note, the Ruby language proved to be a good solution for this project and JRuby also proved to have reached a good maturity level and, in this particular case, no technical problems were detected.

Further developments in this research project shall focus in three distinct areas: use real soccer data, improve event detection and generalization of this soccer heuristics for others CSG. Regarding the first topic, one shall mention the possibility of using this project with a tracking system, allowing the authors to collect and use real soccer data.

Table 5 Accuracy results for goals and outsides

	Goal				Outside			
	Percentage detected	Average of goals observed	Difference	False positive	Percentage detected	Average of outsides observed	Difference	False positive
High goal difference (4)	100	10.75	0	0	92.73	13.75	1	0.25
Small goal difference (5)	100	1.8	0	0	92.11	15.2	1.2	0.4
Draw (3)	100	0.67	0	0	95.65	23	1	0.33
Average (12 games)	100	4.5	0	0	93.20	16.67	1.08	0.33
Standard deviation	0	6.36	0	0	5.40	7.09	0.90	0.49

Table 6 Accuracy results for offsides and intercepted offsides

	Intercepted offside				Offside			
	Percentage detected	Average of intercepted offsides observed	Difference	False positive	Percentage detected	Average of offsides observed	Difference	False positive
High goal difference (4)	100	2.25	0	0	100	2.75	0	0
Small goal difference (5)	93.33	3	0.2	0.2	90	4.2	0.4	0.2
Draw (3)	100	0.67	0	0	88	2.67	0.33	0
Average (12 games)	97.22	2.17	0.08	0.08	93	3.33	0.25	0.08
Standard deviation	3.21	2.76	0.29	0.288	15	3.47	0.62	0.29

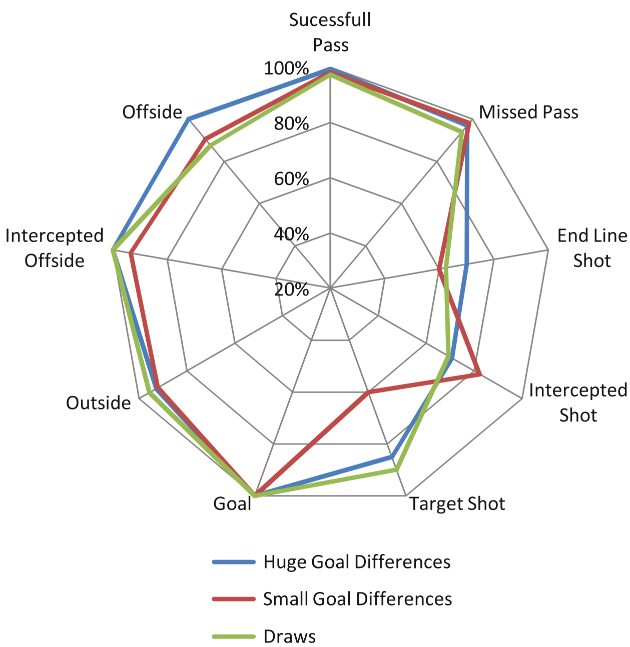


Fig. 6 Comparison between the three groups of games analyzed

Nowadays, there are many different types of tracking systems: camera surveillance system (CSS), radio frequency identification (RFID) or Wi-Fi based, among others, which are capable of covering an outdoor area, like a soccer field, and produce real world data, being CSS methods the most popular ones. This type of solutions are present in many scenarios like public buildings (museums, hospitals, public schools, etc.), private company buildings, and others, focusing mainly on security issues (Zhao and Cheung 2007).

Recently, some CSS solutions were installed in soccer fields in order to track players in a soccer match (Iwase and Saito 2004). In spite of this technology having achieved satisfactory results in the past, some technical problems still remain, like high computer demands and occlusion problems (Khatoonabadi and Rahmati 2009).

Using radio waves as a basis, RFID technology is capable of tracking a person or an object in a specific field. In the acquisition data process, the object being tracked must use a tag, which communicates with a receiver. In spite of these approaches having many applications in several areas, the high costs of both receiver and active tags still constitutes a problem (Chao et al. 2007). Wi-Fi is the standard wireless technology used in many networks, like a university campus or a hospital wireless network. Using this technology as a basis, it is easy to build an infrastructure on top of it, capable of monitoring an object in a specific field using signal triangulation (Abreu et al. 2008). Comparing these three types of approach, it is easy to conclude that RFID and Wi-Fi approaches could constitute good solutions in the soccer area, but since the process of tracking the ball using a tracking system other than CSS still face some technical and regulatory problems, mainly due to the difficulty in introducing a chip in the ball without changing its motion in the field, a good tracking solution should be a hybrid system constituted by a CSS plus an RFID or a Wi-Fi system (the CSS being used to track the ball, and the other system to track the players). Another important issue when using a tracking system is the filtering of the collected data. In spite of the fact that normally a tracking system allows for the definition of a periodicity for information retrieval from the tags, it is very common not to obtain data from all active tags in each period. This point is crucial for the correct detection of the ball and players' positions during the game. Because of that Xavier et al. (2011), developed a system that receives data from a tracking system based on RFID technology, and is capable of reducing the noise present in the raw collected data, using a *low pass* filter. The initial results were very promising and the mean square error was improved by 38% and the maximum error has been reduced by more than 7 cm. The next step consists in expanding this experience to a soccer field and using Kalman and particle filters to improve the results.

Concerning the second topic, by using real soccer data, even though the third dimensional coordinate would be introduced (even though noise can be generated by the localization system, it is also present in the soccerserver which will not constitute a significant obstacle), the authors believe the results would most likely improve, due to the

the lack of abnormal situations, such as the ones mentioned in the previous section. Also, in the future, more events could be detected, especially those related to game breaks, which may be the result of players' injuries, faults marked by the referee, or even an invasion of a sport's fan in the field. Although today some real soccer games have a significant break time, when compared to actual play time, in the 2D robot soccer, there are few of these situations (in average, <2 occurrences per match), so in this work this type of events were not treated. Another feature that could be implemented in the future is related to player behavior throughout a match. In spite of being important, from a coach's perspective, to know what is the opponent player with more shots on target, or the one caught more times in offside, it is extremely important to detect/know the individual and collective behavior patterns, including set plays, of the opponents players. With this type of information, a coach could better prepare his team for a possible match with a specific opponent. Today, there are more than 200 CSG in the world. Over the years, many authors tried to classify these games according to their specific characteristics. For Hughes and Bartlett (2002), collective games can be divided in three distinct groups: games with net and wall, invasion games and games to catch the ball. In this classification, soccer is located in the invasion games group, in the subgroup of games with goals. In this subgroup, another sport is present: hockey. As the CSG universe is very large, one possible feature to be implemented in the future is to expand the soccer heuristics in order to calculate hockey game statistics. When this goal is achieved, a new objective will be expanding the heuristics to other groups of CSG.

Acknowledgements The first and third authors are supported by FCT—Fundação para a Ciência e Tecnologia under grant SFRH/BD/44663/2008 and SFRH/BD/36610/2007, respectively.

References

Abreu P, Costa I, Castelã D, Reis LP, Garganta J (2010) Human vs. robot soccer: how far are they? A statistical comparison. In: Proceedings of the RoboCup symposium 2010. Springer, Berlin, pp 242–253

Abreu P, Vinhas V, Mendes P (2008) Business intelligence through real-time tracking—using a location system towards behaviour pattern extraction. In: Proceedings of the international conference on e-business 2008 (ICE-B 2008), pp 51–57

Acar MF, Yapicioglu B, Arikan N, Yalcin S, Ates N, Ergun M (2008) Analysis of goals scored in the 2006 world cup. In: Reilly T, Korkusuz F (eds) Science and football VI: the proceedings of the sixth world congress on science and football. Taylor & Francis, London, pp 235–241

André E, Binsted K, Tanaka-ishii K, Luke S, Herzog G, Rist T (2000) Three robocup simulation league commentator systems. AI Mag 21(1):57–66

Andre E, Herzog G, Rist T (1988) On the simultaneous interpretation of real world image sequences and their natural language description: the system soccer. In: Proceedings of the 8th European conference on artificial intelligence (ECAI-88), pp 449–454

Binsted K, Luke S, Building AVW (1998) Character design for soccer commentary. In: Asada M, Kitano H (eds) RoboCup-98: robot soccer world cup II. Springer, Berlin, pp 23–35

Carling C, Reilly T, Williams AM (2009) Performance assessment for field sports, 1st edn. Routledge, Taylor & Francis, New York

Carling C, Williams AM, Reilly T (2007) Handbook of soccer match analysis—a systematic approach to improving performance, 2nd edn. Routledge, Taylor & Francis, New York

Chao C-C, Yang J-M, Jen W-Y (2007) Determining technology trends and forecasts of rfid by a historical review and bibliometric analysis from 1991 to 2005. Technovation 27(5):268–279

Chen M, Dorer K, Foroughi E, Heintz F, Huangy Z, Kapetanakis S, Kostiadis K, Kummeneje J, Murray J, Noda I, Obst O, Riley P, Steffens T, Wangy Y, Yiny X (2003) RoboCup soccer server for soccer server version 7.07 and later. The RoboCup Federation

Crampes M, Veuillez JP, Ranwez S (1998) Adaptive narrative abstraction. In: HYPERTEXT '98: proceedings of the ninth ACM conference on hypertext and hypermedia: links, objects, time and space—structure in hypermedia systems. ACM, New York, pp 97–105

Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) Gate: a framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th anniversary meeting of the association for computational linguistics, pp 168–175

Dublin Core Metadata Initiative (1999) Dublin core metadata element set, version 1.1

Ekin A, Tekalp M (2003) Generic play-break event detection for summarization and hierarchical sports video analysis. In: ICME '03: proceedings of the 2003 international conference on multimedia and expo. IEEE Computer Society, Washington, pp 169–172

Frank IM, Miller G (1991) Training coaches to observe and remember. J Sports Sci 9:285–297

Franks IM (1996) Use of feedback by coaches and players. In: Bangsbo J, Reilly T, Williams AM (eds) Science and football III: the proceedings of the third world congress on science and football. Taylor & Francis, London, pp 267–278

Gonzalez I, Abreu P, Reis LP (2008) Using a datawarehouse to extract knowledge from robocup teams. In: Proceedings of ICEIS 2008, pp 511–514

Grehaigne J-F, Bouthier D, David B (1997) Dynamic-system analysis of opponent relationships in collective actions in soccer. J Sports Sci 15:137–149

Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum Comput Stud 43(5–6):907–928

Guarino N (1998) Formal ontology and information systems. In: Proceedings of the first international conference on formal ontology in information systems (FOIS'98), 6–8 June 1998, Trento, Italy. IOS Press, Amsterdam, pp 3–15

Hughes M, Bartlett R (2002) The use of performance indicators in performance analysis. J Sports Sci 10:739–754

Hughes M, Evans S, Wells J (2001) Establishing normative profiles in performance analysis. Perform Anal Sport 1:4–27

Iwase S, Saito H (2004) Parallel tracking of all soccer players by integrating detected positions in multiple view images. In: Proceedings of the 17th international conference on pattern recognition (ICPR '04), vol 4. IEEE Computer Society, Washington, pp 751–754

Khatoonabadi SH, Rahmati M (2009) Automatic soccer players tracking in goal scenes by camera motion elimination. Image Vis Comput 27(4):469–479

Kitano H (ed) (1998a) RoboCup-97: robot soccer world cup I. Springer, London

Kitano H, Asada M, Kuniyoshi Y, Noda I, Osawa E, Matsubara H (1998b) Robocup: a challenge problem for ai and robotics. In: RoboCup-97: robot soccer world cup I. Springer, London, pp 1–19

Ledezma A, Aler R, Sanchis A, Borrajo D (2004) Predicting opponent actions by observation. In: RoboCup 2004: robot soccer world cup VIII. Springer, Berlin, pp 286–296

McGuinness DL (1998) Ontological issues for knowledge-enhanced search. In: Proceedings of the first international conference on formal ontology in information systems (FOIS'98), pp 302–316

McGuinness DL (2003) Ontologies come of age. MIT Press, Cambridge, pp 171–196

Popov B, Kiryakov A, Kirilov A, Manov D, Goranov OM (2003) Semantic annotation platform. In: Proceedings of the 2nd international semantic web conference 2003. Springer, Berlin, pp 834–849

Reis LP, Lau N (2002) Coach unilang—a standard language for coaching a (robo)soccer team. In: RoboCup 2001: robot soccer world cup V. Springer, London, pp 183–192

Rozinat A, Zickler S, Veloso M, van der Aalst W, McMillen C (2009) Analyzing multi-agent activity logs using process mining techniques. In: Distributed autonomous robotic system 8. Springer, Berlin, p 251

Sklansky J (1982) Finding the convex hull of a simple polygon. Pattern Recognit Lett 1:79–83

Smith JR (2001) Mpeg-7 multimedia description schemes. IEEE Trans Circuits Syst Video Technol 11:748–759

Stone P, Veloso M (2000) Layered learning. In: de Mántaras RL, Plaza E (eds) Proceedings of the eleventh European conference on machine learning (ECML 2000). Springer, Barcelona, pp 369–381

Tanaka-Ishii K, Noda I, Frank I, Nakashima H, Hasida K, Matsubara H (1998) Mike: an automatic commentary system for soccer. In: ICMAS'98: proceedings of the 3rd international conference on multi agent systems. IEEE Computer Society, Washington, pp 285–292

Tovinkere V, Qian RJ (2001) Detecting semantic events in soccer games: towards a complete solution. In: IEEE international conference on multimedia and expo, pp 1040–1043

Tsinaraki C, Fatourou E, Christodoulakis S (2003) An ontology-driven framework for the management of semantic metadata describing audiovisual information. In: Eder J, Missikoff M (eds) Proceedings of the 15th international conference on advanced information systems engineering (CAiSE 2003), Klagenfurt, Austria, 16–18 June 2003. Springer, Berlin, pp 340–356

Xavier J, Petry M, Abreu PH, Reis LP (2011) Location and automatic calculation of trend in mobile environments using rfid. In: Proceedings of the Iberian conference of information technologies and systems

Xie L, Xu P, Chang S-F, Divakaran A, Sun H (2004) Structure analysis of soccer video with domain knowledge and hidden markov models. Pattern Recognit Lett 25(7):767–775

Xu G, Ma Y-F, Zhang H-J, Yang S (2003) A hmm based semantic analysis framework for sports game event detection. In: International conference on image processing 2003 (ICIP'03), pp 25–28

Xu P, Xie L, fu Chang S (2001) Algorithms and system for segmentation and structure analysis in soccer video. In: Proceedings of the IEEE international conference on multimedia and expo (ICME), pp 928–931

Yow D, Yeo B, Yeung M, Liu B (1995) Analysis and presentation of soccer highlights from digital video. In: Asian conference on computer vision, pp 499–503

Yu J-q, He Y-f, Sun K, Wang Z-f, Wu X-m (2006) Semantic analysis and retrieval of sports video. In: FCST '06: proceedings of the Japan-China joint workshop on frontier of computer science and technology. IEEE Computer Society, Washington, pp 97–108

Zhao J, Cheung S (2007) Multi-camera surveillance with visual tagging and generic camera placement. In: First ACM/IEEE international conference on distributed smart cameras, 2007 (ICDSC '07), pp 259–266