Annales Mathematicae et Informaticae
46 (2016) pp. 13–36
http://ami.ektf.hu

Avatar-Based Sport Science Soccer Simulations

Norbert Bátfai[ab], András Mamenyák[ab], Péter Jeszenszky[ab], Gergely Kövér[ab], Máté Smajda[ab], Renátó Besenczi[ab], Béla Halász[a], György Terdik[ab], Márton Ispány[ab]

[a]SziMe3D Ltd., Debrecen, Hungary

[b]Department of Information Technology, University of Debrecen

Submitted November 2, 2015 — Accepted October 11, 2016

Abstract

In this paper, we propose a general framework for sport science simulations that we refer to as Simulation Oriented Architecture (SimOA). As a concrete implementation of the framework we present a collection of sport science simulators that we developed in the experimental industrial research and development project called "Football Avatar" for simulating soccer matches. The practical goal of performing such simulations is to help soccer teams with providing an effective tool for supporting tactical decision making. The paper also establishes a solid theoretical foundation for performing sport science simulations introducing the concept of avatars.

Keywords: Forecasting, Parallel Computing, Simulation, Soccer, Sports

MSC: 68U20

1. Introduction

Simulation techniques have been adopted in many fields of science for investigating the behavior of a real system by imitating it through connected artificial objects that exhibit a nearly identical behavior, at least in statistical sense, see the comprehensive book [20]. Simulation has also been contributed significantly to the progress of science, see [7] and provides an important methodological tool in information system research, see [31]. Simulation studies are considered particularly useful for complex systems which cannot be described by simple physical laws, mainly for understanding the behavior of humans on various domains of society, e.g., health care [24], economics and management [9], transportation [30], and sport [26]. The broad scope of the literature in which simulation is applied on these and other fields indicates its relevance.

The research gap to be addressed by this work is to provide a framework for sport science simulations and to establish solid theoretic foundations for that purpose. In order to realize these goals, in the framework of the FootballAvatar project [32] we have developed a novel collection of sport science soccer simulators. This paper presents some key aspects behind these simulators. The simulation algorithms and the technologies used in their implementations will be presented briefly. We call our approach for developing soccer simulators Simulation Oriented Architecture (SimOA) because the design of the developed software system is entirely organized around the logic of simulations. Although our work mainly focuses on soccer, we think that SimOA can be considered as a general approach for performing sport science simulations.

The main purpose of the research presented in this paper was to create a mathematical definition of avatars for sport simulations that was intuitively introduced in [1]. In the intuitive sense, football avatars are computerized abstractions of soccer players, coaches, and referees. However, we will see that the concept of avatars is not limited to be used for soccer only, it can be applied in the case of other sports, among others, ping-pong and tennis. The most important restriction on the simulations is that relevant probability properties both in the simulations and in reality must be the same. One of the most important results of the research is that we were able to develop such simulators.

The paper is organized as follows. The next section gives a general overview of soccer simulation algorithms. The third section introduces definitions for the concept of avatar for soccer simulations and illustrates them with examples. The fourth section discusses our soccer simulation algorithms in detail. The fifth section is dedicated to avatar transformations, biological and behavioural models incorporated into our simulations are also discussed here. Simulation computation results are presented in the sixth section. Finally, the seventh section concludes the paper.

2. Soccer Simulation Computations

2.1. Types of Simulation Models

Paper [4] presents a review of existing soccer simulation models and classifies them in the following three categories: (1) non-realistic, (2) quasi-realistic, (3) and realistic models. Simulation models classified in the first two classes are used in our research on sport science simulations as well as in software components of the FootballAvatar project. In the case of non-realistic models, we have typically used statistical simulation computations. Our core simulation computations are quasi-realistic where statistical methods are mainly used for validating simulation algorithms.

2.2. FootballAvatar Soccer Simulator Collection

The FootballAvatar Soccer Simulation Collection is based on the Simulation Oriented Architecture. The following three levels (or speeds) can be distinguished in the FootballAvatar Soccer Simulation Collection: (1) Level "-1" uses only publicly available or estimated data based on objective and/or subjective observations. (2) Level "0" uses dedicated equipment such as video cameras and sensors to gather data (this infrastructure is provided and operated by our partners). (3) Level "$+$" is built on the lower levels (see Sections 5.2. and 6.2.).

Software elements on each level can operate in the following three modes: (1) The standalone mode does not require any input at all. It is mainly used to generate test data. (2) The analyzer mode serves for analyzing test data or real soccer data. (3) The avatar simulation mode is the basis of the comparison of real and simulated soccer matches, it highly depends on real soccer data.

2.2.1. Statistical Simulation Computations

On level "-1", prediction tasks and related techniques that emerge in football science can be classified into the following classes:

- Predicting the outcomes of an indicator event or an event with a very small number of outcomes. The related techniques are logit, probit and other binary or ordered discrete regression models that contain different explanatory variables, see [14] and [12]. These models can be applied, e.g., to restrict forecasting directly to the match result, i.e., win, draw, or lose.

- Predicting a count type event like the number of faults, goals, or corners. The related techniques are Poisson distribution based models, general bivariate discrete distribution models using copulas, and different algorithms from the field of machine learning.

- Predicting a continuous variable such as the distance covered by players or the time of the possession of the ball by a team during a match. The related techniques are the standard methods of supervised learning to be used for a continuous target such as nonlinear regression and neural networks.

These are typically used in non-realistic simulations.

In Poisson distribution based data analysis, the dependent variable has one- or bivariate Poisson distribution, see [21] and [18]. This framework has been extended in [29] for the time-varying case. A possible application of these models is to forecast the number of goals at a match by bivariate Poisson regression. Poisson distribution based models were criticized in [10] from a football betting market perspective. More general models which use general bivariate discrete distributions generated by copulas have been developed in [23]. In [13], a comparison of goal-driven and ordered regression models can be found.

Machine learning techniques have also been proposed for the prediction of the outcomes of soccer events. In [34], a genetic programming based technique is applied and compared to other two methods based on fuzzy models and neural networks. Applicability of fuzzy rules is also investigated in [28] where the rules are generated by a combination of genetic and neural optimization techniques. More recently, a novel technique, the Bayesian network, which is a graphical probabilistic model, was introduced into soccer science in [8]. A Bayesian network represents the conditional dependencies among uncertain variables, which can be both objective and subjective.

In our competitive programming setup these statistical and machine learning models are competing with each other and they are compared by assessing the quality of their forecasting performance. In the literature, there are various ways for doing assessment, for example, different types of indicators can be considered such as accuracy and profitability. One of the most popular scores is the Rank Probability Score (RPS), see [11] and [8]. In the FootballAvatar system, several objective and subjective goodness-of-fit indicators can be used for assessing the accuracy of the forecasts derived by models mentioned above. In particular, it is also possible to compare our prediction with bookmakers' ones.

2.2.2. Core Simulation Computations

From the viewpoint of implementations, we distinguish the following two main types of simulators: the MABSA ones and the FANM ones. While MABSA (Multi-Agent-Based Server Architecture) is used for research purposes only, commercial software components of the FootballAvatar project are based on our FANM (FANM is Not MABSA) platform. FANM is the antithesis of MABSA. For example, the heart of the MABSA platform is an asynchronous I/O multiplexed TCP/IP proxy server written in C++11. In the MABSA platform teams, players, coaches, 2D, 3D and mobile display programs and the simulation algorithms themselves are implemented as clients that communicate with the server via TCP/IP using Google's Protobuf [15]. Conversely, FANM programs are standalone monolithic applications that do not use networking at all.

On higher levels, MABSA and FANM simulations are typically quasi-realistic ones. In contrast to realistic simulations (such as 2D robot soccer [19], simplified 2D robot soccer [3], or Simple Soccer [6, p. 133–193]), our quasi-realistic computations are organized around a few key features of the soccer game, such as passing graphs or lineups. The base algorithms of MABSA and FANM as well as their software infrastructure are presented in Section 4. More advanced simulation models such as the ones that use cellular networks or Bayesian networks will be discussed in a further paper.

2.3. Competitive Programming

During the development of the FootballAvatar project we have developed a new software process methodology that we call Competitive Programming (or CP for short) [4]. This methodology is based on a combination of eXtreme Programming (XP) ([5], [36]) and Rapid Application Development (RAD) [22]. Simulation programs presented in this paper were developed according to CP.

3. Definitions of Avatars

As the main result of this paper, the mathematical and the information technological definitions of avatars are presented in this section.

3.1. Mathematical Definition

The spirit of the following statistics-based definition may be deduced from the hypothesis testing of [2]. In addition, a heuristic version of this definition can be found in the paper [4].

First, let us select n number of properties that will be observed in simulations. These quantities are arranged in an n-dimensional random variable $\mathbf{X} = (X_1, \dots, X_n)$. The realizations of this vector $\mathbf{X}$ will be called a-priori observations with a-priori distribution function F. For example, X_1 may be the number of goals scored by a given team in a given soccer match. This, of course, depends on chance but it has well-known realizations from the past. Simulations will give further realizations of $\mathbf{X}$ and all we have to do is to compare these realizations to a-priori observations. This approach is supported by the following definition of avatars.

Definition 3.1 (Avatars)**.** Let $\mathbf{X} = (X_1, \dots, X_n)$ be the selected properties and let $\mathcal{S}\colon U \to \mathbb{R}^n$ symbolize a simulation algorithm, where U denotes an arbitrary (possibly empty) set of inputs of this algorithm. The pair $(\mathbf{X}, S)$ is referred to as an avatar (with significance level α) if the null hypothesis $H_0 : F = F_S$ is not being rejected, where F_S denotes the distribution function of the random outputs of the simulation algorithm.

It is possible to weaken this definition in the form that the hypotheses $H_0 : F_i = F_{S,i}$ are investigated for all $i = 1, \dots, n$, where F_i and $F_{S,i}$ denote the distribution functions of the ith avatar property and the ith output coordinate of the simulation algorithm, respectively. The difference between the two definitions is that in the first case the joint distributions are compared, while in the second case only the marginal ones.

If the simulation has no input at all or its input is not decisive then it is referred to as standalone, otherwise it is called avatar-based. Here we present some trivial thought experiments with the definition, and a few non-trivial ones can be found in Section 6.2.

Example 3.2 (A trivial ping-pong avatar)**.** Suppose that we observed two table tennis players during their last 10 matches. In each game, the relative frequencies of their successful serves and returns were determined. Let $y^{1,1}$, $y^{1,2}$, $y^{2,1}$, and $y^{2,2}$ denote the vectors of these relative frequencies. To be more precise, the sequence $y^{p,j} \colon \{1, \dots, 10\} \to [0,1]$ tells us the relative frequency of successful serves ($j = 1$) and returns ($j = 2$) of player p ($p = 1, 2$) in the ith game ($i = 1, \dots, 10$).

Let assume that we have the following observations:

$$\begin{aligned}
y^{11} &= (21/41, 10/36, 23/40, 15/28, 19/32, 10/33, 24/42, 12/41, 12/30, 29/33),\\
y^{12} &= (11/25, 16/21, 6/21, 15/16, 12/29, 14/23, 10/19, 7/15, 6/14, 6/16),\\
y^{21} &= (11/41, 17/30, 14/23, 11/31, 14/22, 8/21, 14/27, 8/17, 11/14, 12/20),\\
y^{22} &= (19/39, 17/28, 12/33, 8/24, 14/24, 10/18, 8/27, 20/30, 11/29, 11/26).
\end{aligned}$$

Now, for example, let us consider the last 5 matches only. Accordingly, let

$$p_{11} = \sum_{i=6}^{10} y_i^{11}/5 = 0.4891859, \quad p_{12} = \sum_{i=6}^{10} y_i^{12}/5 = 0.4810499,$$
$$p_{21} = \sum_{i=6}^{10} y_i^{21}/5 = 0.5511547, \quad p_{22} = \sum_{i=6}^{10} y_i^{22}/5 = 0.4641812.$$

Then the a-priori probabilities can be estimated as

$$\mathbf{X} = \begin{pmatrix} \text{player 1's serves} \\ \text{player 1's returns} \\ \text{player 2's serves} \\ \text{player 2's returns} \end{pmatrix} = \begin{pmatrix} \frac{p_{11}}{p_{11}+p_{22}} \\ \frac{p_{12}}{p_{12}+p_{21}} \\ \frac{p_{21}}{p_{21}+p_{12}} \\ \frac{p_{22}}{p_{22}+p_{11}} \end{pmatrix} = \begin{pmatrix} 0.5131139 \\ 0.4660412 \\ 0.5339588 \\ 0.4868861 \end{pmatrix}.$$

The components of this vector denote the estimated probabilities of successful serves and returns. To be more precise, they indicate whether the serving or the returning player gets the point. We define the avatar data transformation function as

$$\mathcal{A}(y^{11}, y^{12}, y^{21}, y^{22}) = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}.$$

$\mathcal{A}$ maps the observations to the input parameters of the algorithm S. The IT implementation of avatar transformations is discussed in Section 5.

For simulations we have used the algorithm S shown in Fig. 2(a). Our practical experience shows that $(\mathbf{X}, S \circ \mathcal{A})$ is an avatar.

Example 3.3 (Tennis avatar)**.** In this example we consider data about the two finalists of the Australian Open 2014 tennis championship, Rafael Nadal and Stanislas Wawrinka available from [33]. In each game of the tournament the relative frequencies of the points won after the 1st serve and the points won after the opponent's 1st serve were collected. Let $y^{1,1}$, $y^{1,2}$, $y^{2,1}$, $y^{2,2}$ denote the vectors of these measurements. To be more precise, the sequence $y^{p,j} \colon \{1, \dots, 7\} \to [0,1]$

tells us the relative frequencies of the points won after the 1st serve ($j = 1$) and the points won after the opponent's first serve ($j = 2$) of the pth player ($p = 1, 2$) in the ith game ($i = 1, \dots, 7$). The first player is Stanislas Wawrinka, the second one is Rafael Nadal.

In our case we have the following:

$$\begin{aligned} y^{11} &= (20/22, 57/69, N/A, 54/60, 71/98, 71/87, 50/84), \\ y^{12} &= (18/36, 25/81, N/A, 23/76, 30/105, 16/88, 7/53), \\ y^{21} &= (14/16, 45/55, 38/53, 67/88, 66/91, 41/56, 46/53), \\ y^{22} &= (5/20, 19/52, 29/68, 25/76, 26/90, 24/69, 34/84). \end{aligned}$$

Note that the 3rd match of Wawrinka was canceled. In the following we consider only the values of the last 4 matches. Accordingly, let

$$p_{11} = \sum_{i=4}^{7} y_i^{11}/4 = 0.85152475, \quad p_{12} = \sum_{i=4}^{7} y_i^{12}/4 = 0.22555988,$$
$$p_{21} = \sum_{i=4}^{7} y_i^{21}/4 = 0.76058081, \quad p_{22} = \sum_{i=4}^{7} y_i^{22}/4 = 0.34260606.$$

Then the a-priori probabilities can be estimated as

$$\mathbf{X} = \begin{pmatrix} \text{Wawrinka's serves} \\ \text{Wawrinka's returns} \\ \text{Nadal's serves} \\ \text{Nadal's returns} \end{pmatrix} = \begin{pmatrix} \frac{p_{11}}{p_{11}+p_{22}} \\ \frac{p_{12}}{p_{12}+p_{21}} \\ \frac{p_{21}}{p_{21}+p_{12}} \\ \frac{p_{22}}{p_{22}+p_{11}} \end{pmatrix} = \begin{pmatrix} 0.71309168 \\ 0.22872992 \\ 0.77127008 \\ 0.28690832 \end{pmatrix}.$$

The components of this vector denote the estimated probabilities of the points won after the 1st serve and the points won after the opponent's first serve. To be more precise, they indicate whether the serving or the returning player gets the point. We define the avatar data transformation function as

$$\mathcal{A}(y^{11}, y^{12}, y^{21}, y^{22}) = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}.$$

For simulations we have used the algorithm S shown in Fig. 2(b) that is analogous with the one used for the ping-pong avatar. Our practical experience shows that $(\mathbf{X}, S \circ \mathcal{A})$ is an avatar.

Example 3.4 (Trivial dribbling-tackling or one-dimensional football)**.** Let us consider two soccer players A and B as it is shown in Fig. 1. During their last 5 matches, these two players were observed in order to determine the relative frequencies of their successful dribbles and tackles. Let y^{AT}, y^{AD}, y^{BT}, and y^{BD} denote the vectors of these relative frequencies. With the notation of Definition 3.1, we have the following:

$$y^{AT} = (3/9, 2/8, 3/10, 4/10, 2/8), \quad p_{AT} = \sum y_i^{AT}/5 = .3067,$$

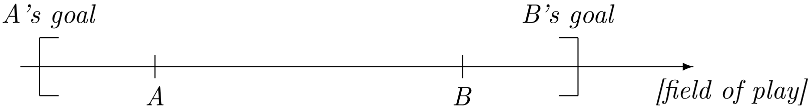


Figure 1: The informal interpretation of the one-dimensional football, where A and B are the two investigated players. For example, a situation $A_t > B_t$ may denote that the attacker A dribbles the defender B (and accordingly, B cannot tackle A), or vice versa, at time t.

$$\begin{aligned}
y^{AD} &= (6/8, 9/10, 6/6, 4/8, 6/7), \quad & p_{AD} &= \sum y_i^{AD}/5 = .8014, \\
y^{BT} &= (8/10, 6/7, 6/6, 8/9, 5/6), \quad & p_{BT} &= \sum y_i^{BT}/5 = .8759, \\
y^{BD} &= (3/10, 2/8, 4/9, 3/5, 1/4), \quad & p_{BD} &= \sum y_i^{BD}/5 = .3689.
\end{aligned}$$

Let's estimate the a-priori probabilities for the experiment by the following:

$$\mathbf{x} = \begin{pmatrix} 1 - \frac{p_{BD}}{p_{BD}+p_{AT}} \\ \frac{p_{AD}}{p_{AD}+p_{BT}} \\ 1 - \frac{p_{AD}}{p_{AD}+p_{BT}} \\ \frac{p_{BD}}{p_{BD}+p_{AT}} \end{pmatrix} = \begin{pmatrix} 0.4539668 \\ 0.4777917 \\ 0.5222083 \\ 0.5460332 \end{pmatrix}.$$

We define the avatar data transformation function as

$$\mathcal{A}(y_t^{AT}, y_t^{AD}, y_t^{BT}, y_t^{BD}) = \begin{pmatrix} p_{AT} & p_{BT} \\ p_{AD} & p_{BD} \end{pmatrix}.$$

The simulation S is implemented by the R code shown in Fig. 2(b).

(a) The R code of the ping-pong avatar example.

```r
serve <- function(p) {
  u <- runif(1)
  if(u < p)
    return(1)
  else
    return(0)
}

p_1s <- 0.5131
p_1r <- 0.4660
p_2s <- 0.5339
p_2r <- 0.4868

nofmatches <- 10000
c_1s <- 0
c_1r <- 0
c_2s <- 0
c_2r <- 0

for(i in 1:nofmatches) {
  for(j in 1:10) {
    if(serve(p_1s)) {
      c_1s <- c_1s + 1
    } else {
      c_2r <- c_2r + 1
    }
    if(serve(p_2s)) {
      c_2s <- c_2s + 1
    } else {
      c_1r <- c_1r + 1
    }
  }
}

cat(c_1s / (nofmatches * 10),
  c_1r / (nofmatches * 10),
  c_2s / (nofmatches * 10),
  c_2r / (nofmatches * 10),
  "\n"
)
```

(b) The R code of the one-dimensional football.

```r
attack <- function(p_d, p_t) {
  p <- p_d * (1 / (p_d + p_t))
  u <- runif(1)
  if(u < p)
    return(1)
  else
    return(0)
}

p_AT <- 0.3067
p_AD <- 0.8014
p_BT <- 0.8759
p_BD <- 0.3689

nofmatches <- 10000
AT <- 0
AD <- 0
BT <- 0
BD <- 0

for(i in 1:nofmatches) {
  for(j in 1:10) {
    if(attack(p_AD, p_BT)) {
      AD <- AD + 1
    } else {
      BT <- BT + 1
    }
    if(attack (p_BD, p_AT)) {
      BD <- BD + 1
    } else {
      AT <- AT + 1
    }
  }
}

cat(AT / (nofmatches * 10),
  AD / (nofmatches * 10),
  BT / (nofmatches * 10),
  BD / (nofmatches * 10),
  "\n"
)
```

Figure 2: The two trivial simulation algorithms of the introductory examples.

The results from running the simulation algorithms in Fig. 2 shows that $(\mathbf{x}, S \circ \mathcal{A})$ is an avatar.

Definition 3.5 (Football avatar)**.** Let $\mathbf{X} = (X_1, \ldots, X_n)$ be the investigated avatar properties. An avatar $(\mathbf{X}, S \circ \mathcal{A})$ is referred to as a passing distribution or lineup-based football avatar if $\mathcal{A} : U \to \mathbb{R}^{(11+k)\times 11}$ and $S : \mathbb{R}^{(11+k)\times 11} \to \mathbb{R}^n$, where k denotes the number of control avatar parameters.

The number 11 corresponds to the starting 11, however, the whole team must be used in practical applications.

Definition 3.6 (Deep avatar)**.** A football avatar $(\mathbf{X}, S \circ \mathcal{A})$ is referred to as a deep avatar with depth v if $\mathcal{A} = \mathcal{A}_v \circ \mathcal{A}_{v-1} \circ \cdots \circ \mathcal{A}_0$, where $\mathcal{A}_0 \colon U \to \mathbb{R}^{(11+k)\times 11}$ and $\mathcal{A}_i \colon \mathbb{R}^{(11+k)\times 11} \to \mathbb{R}^{(11+k)\times 11}$, for $i = 1, \ldots, v$.

Remark 3.7 (Simulational, molecular, physiological, mechanical, and psychological avatar data transformations)*.* Let $(\mathbf{X}, S \circ \mathcal{A})$ be a deep avatar with $\mathcal{A} = \mathcal{A}_4 \circ \mathcal{A}_3 \circ \mathcal{A}_2 \circ \mathcal{A}_1 \circ \mathcal{A}_0$, where $\mathcal{A}_0$ is called a simulational (or tactical), $\mathcal{A}_1$ is called a molecular, $\mathcal{A}_2$ is called a physiological, $\mathcal{A}_3$ is called a mechanical, and finally, $\mathcal{A}_4$ is called a psychological avatar data transformation function. It is interesting to notice that the simulational avatar transformation $\mathcal{A}_0$ is a strongly distinguished function in the definition of deep avatars.

3.2. Information Technological Definition

In the sense of information technology, football avatars are simply cross cutting concepts between tactical avatars and higher order avatars. For example, it is obvious that players' passing probabilities or shooting accuracies are influenced by their physiological and psychological states. The basic data of tactical avatars are based on the passing matrices, lineups, and some other avatar control properties. These empirical and/or predicted quantities are used by our simulation software by default. Therefore, it is natural to use aspects (in the sense of Aspect Oriented Programming) to implement the influences of several internal and external factors to the tactical avatars. For example, external factors to be taken into consideration include environmental aspects (such as weather conditions and properties of the pitch) or the referee.

Example 3.8 (A top goalscorer aspect)**.** Suppose that we have tactical avatars for both the next round's home and away teams, but the home team has already won the national championship, so it can play without stakes. However, let us assume that it is possible for a striker of the home side to win the top goalscorer title. Accordingly, the tactic of the home team has been changed to achieve this goal. It is an interesting question how this change impacts the tactical avatars for the simulation of the next match. For example, it is necessary to modify the passing distribution matrix and the selfishness control property. It can be done quickly and easily by writing an appropriate aspect.

4. Soccer Simulation Algorithms

4.1. Statistical Simulation Computations

This kind of simulation computation is typically used on level "-1". In this paper, the bivariate Poisson regression model is described in more detail. In this model, the result of a match, which is a pair of the number of goals scored by the home and the away teams (denoted by X and Y, respectively), has a bivariate Poisson distribution. This distribution is a linear transformation of three independent Poisson distributions. Namely, (X, Y) follows the bivariate Poisson distribution with parameters $\lambda_1, \lambda_2, \lambda_3$, if $X = Z_1 + Z_3$ and $Y = Z_2 + Z_3$, where Z_1, Z_2, Z_3 follow independent Poisson distributions with parameters $\lambda_1, \lambda_2, \lambda_3$, respectively. Clearly, X and Y are correlated with the correlation coefficient λ_3. In our case, X and Y denote the number of goals scored by the home and away teams, respectively, while Z_3 denotes the number of goals scored by both teams, Z_1 and Z_2 are the goal differences by which the match was won and lost, respectively, from the viewpoint of the home team. In the bivariate Poisson regression model, the intensity parameters $\lambda_1, \lambda_2, \lambda_3$ depend on various covariates, e.g., the team indicators or the home playing indicator. The parameters incorporated by these covariates into the model are able to measure the impact of the home pitch or the team strength at home or away.

Algorithm 1 The EM algorithm for computing team scores.

```
Require: $match\_results$ : array of $(home, away)$ team results and $design\_matrices$ : $X_w, X_l, X_d$; ▷ The win, lose, and draw design matrices are defined by team-indicators and additional covariates.
Ensure: $parameters$ : $\beta_w, \beta_l, \beta_d$; ▷ The parameters according to winning, losing, and drawing of the bivariate Poisson distribution.
 1: procedure TeamScore($match\_results$, $design\_matrices$, $intensities$)
    ▷ Initialization
 2:     $g_d \leftarrow \min\{home, away\}$;                ▷ Goals by both teams.
 3:     $g_w \leftarrow home - g_d$;                        ▷ Winning goals.
 4:     $g_l \leftarrow away - g_d$;                        ▷ Losing goals.
 5:     $\beta_w \leftarrow$ glm($X_w, g_w$, "poisson");  ▷ Fitting Poisson regression for winning goals by using the glm procedure.
 6:     $\beta_l \leftarrow$ glm($X_l, g_l$, "poisson");  ▷ Same for losing goals.
 7:     $\beta_d \leftarrow$ glm($X_d, g_d$, "poisson");  ▷ Same for draw.
 8:     $\lambda_w \leftarrow \exp\{X_w * \beta_w\}$;      ▷ Computing initial winning intensity.
 9:     $\lambda_l \leftarrow \exp\{X_l * \beta_l\}$;      ▷ Computing initial losing intensity.
10:     $\lambda_d \leftarrow \exp\{X_d * \beta_d\}$;      ▷ Computing initial draw intensity.
11:     repeat
    ▷ Expectation step
12:         $\varrho \leftarrow \lambda_d/(\lambda_w * \lambda_l)$;
13:         if $(home > 0) \& (away > 0)$ then
14:             $ratio \leftarrow G[home, away]/G[home + 1, away + 1]$;   ▷ $G$ is computed by CondExp
15:             $g_d \leftarrow home * away * \varrho * ratio$;
16:         else
17:             $g_d \leftarrow 0$;
18:         end if
19:         Repeat steps 3 and 4 with new $g_d$;
    ▷ Maximization step
20:         Repeat steps 5, 6, and 7 with new $g_w, g_l, g_d$;
21:         Repeat steps 8, 9, and 10 with new $\beta_w, \beta_l, \beta_d$;
22:     until convergence;
23:     return $\lambda_w, \lambda_l, \lambda_d$;
24: end procedure
```

The parameters of bivariate Poisson regression are estimated by an EM-type algorithm, see Algorithm 1. This algorithm consists of two steps: the expectation (E) and the maximization (M) steps. In the E-step, the independent components Z_1, Z_2, Z_3 are estimated from marginals X and Y using conditional expectation, see Algorithm 2. In the M-step, three separate Poisson regressions are fitted using the generalized linear model (GLM) framework. For example, this fitting can be done by using the `family = poisson` option in the `glm` function of R. The output of Algorithm 1 can be, for example, the home, away, and draw strength of the teams in a league which can be shown as in Fig. 4 in the case of the Hungarian

National Championship during season 2011/2012.

Algorithm 2 Computing the G matrix in Algorithm 1.

```
Require: $\varrho$ : real correlation, $max\_goal$ : integer maximum number of goals;
Ensure: $G : (max\_goal + 1) \times (max\_goal + 1)$ array;
procedure CondExp($\varrho$,$max\_goal$,$G$)
    $G \leftarrow 1$;                                                  ▷ Initialization
    for $j = 1$ to $max\_goal$ do
        for $k = 1$ to $j$ do
            $G[j+1, k+1] \leftarrow G[j+1, k] + (j+1) * \varrho * G[j, k]$;
            $G[k+1, j+1] \leftarrow G[j+1, k+1]$;
        end for
    end for
    return $G$;
end procedure
```

4.2. Core and Massively Parallel Simulation Computations

Our core soccer simulation algorithms are implemented as quasi-realistic MABSA or FANM simulation computations. The two most successful MABSA soccer algorithms were *FBA One* and *TF*. Their pseudo-codes are shown in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 The default simulation algorithm used by the team FBA One.

```
Require: ActorRequest $actors[]$;      ▷ The class ActorRequest is a Google Protocol Buffers object.
Ensure: SensoryInput $response$;       ▷ The class SensoryInput is a Google Protocol Buffers object.
procedure A simulation step($actors[]$)                              ▷ Called at discrete time step $t$
    $t \in \mathbb{N}$ is the discrete time index, Players $players[]$, Ball $ball$;   ▷ Server side global objects.
    Player $p$ = possession of the ball();                   ▷ Who has the possession of the ball?
    if SHOOT or PASS $\in actors[p]$ then
        start ball moving simulation();   ▷ Runge-Kutta to predict the future motion of the ball.
    end if
    moving(ball);
    for each $q \in players$ do
        moving(q);
    end for
    Searching for actions or play modes();
    response = create response(ball, actors, players);
    return $response$;                              ▷ This will be the sensory input of the players.
end procedure
```

FANM algorithms were developed from the successful MABSA algorithms. The architecture of FANM teams are suitable for porting these to GPU in order to implement massively parallel simulation computations. All CUDA ported versions are shown in Fig. 3. Two of these ports will be presented in detail in Sect. 6.3.

5. Avatar Transformations

5.1. Soccer Analytics

In order to use a soccer simulator in practice it should be driven by real soccer data. Our simulation programs developed in the FootballAvatar project also provide such

Algorithm 4 The simulation algorithm used by the team TF.

```
Require: ActorRequest actorRequest[];
Ensure: SensoryInput sensoryInput;    ▷ The classes ActorRequest and SensoryInput are Google
    Protocol Buffers objects.
procedure SIMULATE(SensoryInput)
    for $i = 0$ to all_players_size do
        player ← mutable_all_players(i)
        switch state_of_play
            case BALL_IN_PLAY
                switch player.action_type
                    case MOVE
                        move_ball_in_play(player)
                    case KICK
                        kick_ball_in_play(player)
                    ...
            ...
    end for

    runge_kutta()                      ▷ Runge-Kutta to predict the motion of the ball.
    time_stamp ← time_stamp + 1

    sleep()
    write_server(SensoryInput)
end procedure
```

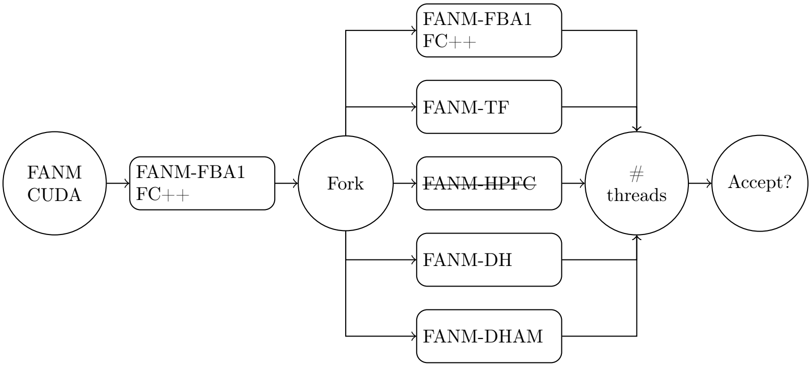


Figure 3: An arrangement of the CUDA ports of prototypes in our competition programming flow chart. The object of the contest is to maximize the number of parallel threads of soccer matches in a CUDA block.

functionality. We have developed file formats for representing soccer data that originate from wearable sensors and from video processing. Our partner companies provide and operate an infrastructure that can supply us with soccer data in our file formats. In order to be useful for our simulators real soccer data is subject to processing, that is referred to as Basic Avatar Transformation (or BAT for short).

The output of BAT serves as input to our simulators. Currently, BAT computes a passing matrix for both teams and another matrix called "heat map" for each player. Each element of a "heat map" matrix corresponds to a specific rectangular region of the pitch (the pitch is divided into disjoint regions by an equidistant grid).

A number in the "heat map" matrix represents the time spent by the player in the corresponding region.

The FANM-Debrecen Handsomes (or FANM-DH for short) simulation algorithm (mentioned later in Sect. 6.2) uses AspectJ crosscutting to manipulate certain parameters. One of these parameters is selfishness. Assume that one of our players, for some reason, shoots on goal with greater probability than he passes in some situations. The crosscutting made for the FANM-DH algoritm can manipulate this probability. By these AspectJ crosscuts we are able to manipulate more player parameters, constructing every players properties.

5.2. Biological and Behavioural Models

The Player Stamina Avatar Property (PSAP) is introduced at higher levels. Each player has a personalized fitness level based on real-life measured data. Basic physical parameters (e.g., weight, height), sport performance indices (e.g., 40 yard dash, fitness tests) and basic physiological/haematological levels (e.g., blood pressure, blood sugar) determine a measure in the range between 1 and 1000 PSL (the abbreviation PSL stands for Player Stamina Level). In simulations the value of this measure is continuously decreased based on the players' behaviour (e.g, running at a higher speed results in more decrease). Because it is practically not possible to obtain all these data before every match, this avatar property have to be estimated based on the matches since the last measurement. PSAP works like the 2DRCSS's stamina model [27], but is more realistic, because it is based on real data. In our plans, a more advanced PSAP will be able to indicate suspicious events like doping, not from measured data like the Union Cycliste Internationale's (UCI) Athlete Biological Passport [35], but from the difference between the reality and the simulations.

The Fouls Avatar Property (FAP) is also introduced at higher levels. Each player varies in how aggressive he/she is in the pitch. However, the level of aggression can be characterized by the number of yellow and red cards awarded, or the number of fouls committed during previous matches. FAP incorporates these factors, as well as some others. For example, aggression is influenced by mood: after making a successful tackle a player is likely to become more aggressive, that implies higher chance of making a foul. Furthermore, FAP also interacts with PSAP, e.g., the more tired a player is, the higher is the probability of committing a mistake.

6. Simulation Computation Results

6.1. Statistical Simulation Computations

The simulation computations used or investigated on level "-1" may typically give several alternative league tables. Now, we compare two families of such league tables, namely the alternative and the PCA-Poisson league tables that are shown in Table 1 together with the real league table for the Hungarian National Championship. The order in the alternative league table is based on Google's Page Rank algorithm [25]. In this case, there is an edge from team A to team B in the Google matrix if team B can score at least one point against team A. An example of an alternative league table can be seen in the middle part of Table 1. Uses of Page Rank in sport performance analysis can be found in the scientific literature, e.g., see [17] and [16]. One of the authors continuously maintain some Hungarian language web pages devoted to alternative league tables at `http://hu.wikipedia.org/wiki/Alternat%C3%ADv_tabella` and `http://nsza.blog.hu/`. This activity takes place independently from the FootballAvatar project and these pages contain alternative league tables for the Premier League, Serie A, Primera División, and the Bundesliga, too. On level "-1" of the FootballAvatar project Page Rank algorithm-based experiments are performed by two competing development teams (see Table 1).

Table 1: The first two columns show the official league table for the Hungarian National Championship for the 2011/2012 season. The third and fourth columns contain an alternative league table for the same season. The fifth and sixth columns contain a PCA-Poisson based league table for the same season as well.

Official		Alternative		PCA-Poisson	
Team	Points	Team	Rank	Team	Score
Debrecen	74	Debrecen	0.0992	Debrecen	1.2021
Videoton	66	Kaposvár	0.0755	Videoton	1.1666
Győr	63	Siófok	0.0719	Győr	0.9545
Honvéd	46	Honvéd	0.0702	Kecskemét	0.3926
Kecskemét	45	Videoton	0.0696	Haladás	0.2676
Paks	45	Győr	0.0682	Diósgyőr	0.0232
Diósgyőr	43	Paks	0.0679	Paks	0.0045
Haladás	38	Haladás	0.0646	Újpest	-0.1163
Siófok	36	Pécs	0.0631	Pécs	-0.1951
Kaposvár	35	Kecskemét	0.0615	Honvéd	-0.2376
Ferencváros	34	Vasas	0.0535	Kaposvár	-0.2856
Pécs	34	Pápa	0.0517	Siófok	-0.3628
Újpest	32	Diósgyőr	0.0499	Ferencváros	-0.4397
Pápa	30	Ferencváros	0.0498	Pápa	-0.4843
Vasas	22	Újpest	0.0473	Vasas	-0.6430
Zalaegerszeg	13	Zalaegerszeg	0.0353	Zalaegerszeg	-1.2467

The other league table called PCA-Poisson is based on the EM algorithm shown in Alg. 1 and Principal Component Analysis (PCA) applied for the three dimensional vector with coordinates home, away, and draw strength. The one-dimensional score given by the dimension reduction technique PCA is shown in the last column of Table 1.

6.2. Core Simulation Computations

Core simulation computations were introduced in [4]. We have collected the total number of points and goals during seasons 2004/2005–2012/2013 in the Hungarian National Championship, that are shown in Table 2. Note that $\overline{x}_g$ and $\overline{x}_p$ denote the mean of goals and points respectively, and s^*_{ng} and s^*_{np} denote the standard

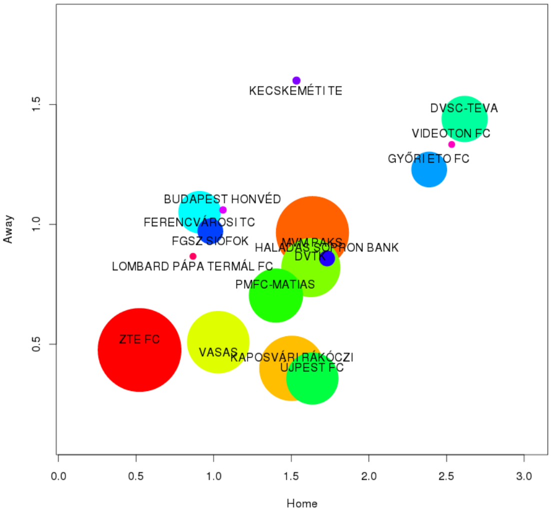


Figure 4: The scatterplot of home and away team strengths with draw strength in bubble size.

deviation of goals and points respectively.

Season	4/5	5/6	6/7	7/8	8/9	9/10	10/11	11/12	12/13
Goals	681	707	677	746	710	707	690	648	639
Points	644	658	671	666	667	659	667	656	660

Table 2: The total number of goals and points scored during seasons 2004/2005–2012/2013 in the Hungarian National Championship ($\overline{x}_g = 689.44$, $s^*_{n\,g} = 33.02$, $\overline{x}_p = 660.89$, and $s^*_{n\,p} = 8.10$).

We have simulated nine seasons for this championship using the FANM-Debrecen HardAsMuscle and the FANM-Debrecen Handsomes algorithms, Tables 3 and 4 shows the total number of goals and points obtained from these simulations. These results were compared with the similar statistical results in [4]. The significance level α for all tests is chosen as $\alpha = 0.05$. Our findings show that both simulation algorithm passes the statistical tests applied.

6.3. Massively Parallel Simulation Computations

In this point, two CUDA ported FANM teams are presented briefly. All FANM and CUDA ported FANM teams have the same functionality because they were inherited from a common ancestor.

Season	1.	2.	3.	4.	5.	6.	7.	8.	9.	Test stat.
Goals	611	626	649	639	596	606	644	610	622	6/78
Points	663	670	672	658	665	667	671	685	669	10/21

Table 3: The total number of goals and points obtained from the simulation algorithm FANM-Debrecen HardAsMuscle ($\overline{x}_g = 622.56$, $s^*_{n\,g} = 18.41$, $\overline{x}_p = 668.89$, and $s^*_{n\,p} = 7.47$). The last column shows the values of test statistics in form of Wald-Wolfowitz/Mann-Whitney.

	1.	2.	3.	4.	5.	6.	7.	8.	9.	Test stat.
Goals	704	674	703	701	714	665	694	679	682	11/42
Points	667	658	669	665	665	657	666	661	656	9/41

Table 4: The total number of goals and points obtained from the simulation algorithm FANM-Debrecen Handsomes ($\overline{x}_g = 690.67$, $s^*_{n\,g} = 15.42$, $\overline{x}_p = 662.67$, and $s^*_{n\,p} = 4.5$). The last column shows the values of test statistics in form of Wald-Wolfowitz/Mann-Whitney.

6.3.1. The FANM-TF CUDA Implementation

This algorithm simulates several thousand matches between two teams. By default, each team has 4 formations (as shown in Table 5) with a corresponding passing distribution and avatar property table. This gives us 16 matchups, each of which has its own CUDA block. The matches are played on separate threads but with common constant data. The algorithm has two kernel launches for each of the two halves with an in-between initialization of the second half.

Lineups	4-4-2	4-2-2-2	4-2-3-1	4-3-3
4-4-2	240/27/133	235/36/129	357/13/30	389/4/7
4-2-2-2	163/23/214	189/35/176	314/16/70	369/6/25
4-2-3-1	78/16/306	102/23/275	219/15/166	319/15/66
4-3-3	15/13/372	31/9/360	58/10/332	195/16/189

Table 5: FANM-TF's Hungarian flag notation.

6.3.2. The FANM-Debrecen Handsomes CUDA Implementation

The results are less predictable compared to FANM-TF CUDA implementation, given that the differences between the players (avatars) of the teams reflected by

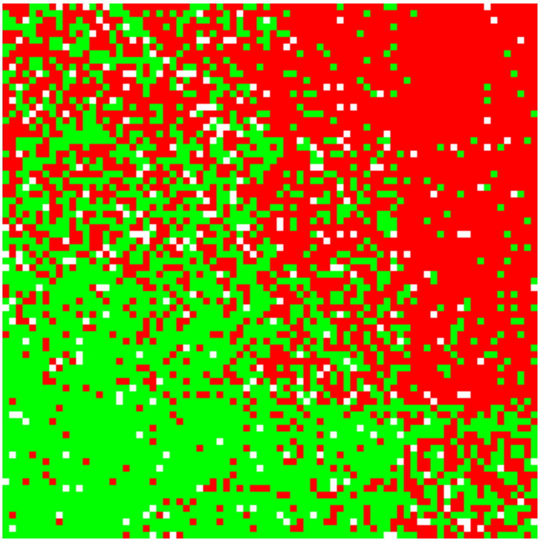


Figure 5: FANM-TF's Hungarian flag notation.

Lineups	4-4-2					4-2-2-2					4-2-3-1					4-3-3				
4-4-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	398	0	0	0	0	400	0	0	0	1	399	0	0	0	1	399
4-2-2-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	400	0	0	1	1	398	0	0	0	0	400	0	0	0	0	400
4-2-3-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	400	0	0	0	0	400	0	0	0	0	400	0	0	0	0	400
4-3-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	400	0	0	0	0	399	0	0	0	0	400	0	0	0	0	400

Table 6: FANM-TF's crystal ball notation.

this algorithms are less significant and the players of a team have quite similar characteristics (see Fig. 6 and Table 8).

6.4. Avatar Simulation Computations

We have simulated nine seasons for the same championship as in Sect. 6.2 using the FANM-TF algorithm. The total number of goals and points obtained from the simulations are shown in Table 9. These results was compared with the similar statistical results in [4]. Our findings show that this simulation algorithm passes

Lineups	4-4-2	4-2-2-2	4-2-3-1	4-3-3
4-4-2	127/148/125	162/88/150	88/119/93	143/113/144
4-2-2-2	158/83/159	140/119/141	186/115/99	136/121/143
4-2-3-1	96/130/174	101/131/168	8/200/102	79/141/180
4-3-3	149/116/135	133/123/144	147/147/106	136/130/134

Table 7: FANM-Debrecen Handsomes' Hungarian flag notation.

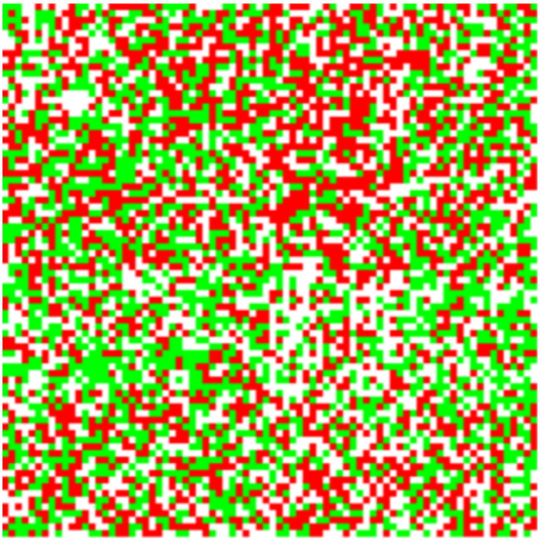


Figure 6: FANM-Debrecen Handsomes' Hungarian flag notation.

<table>
<tr><th>Lineups</th><th>4-4-2</th><th>4-2-2-2</th><th>4-2-3-1</th><th>4-3-3</th></tr>
<tr><td>4-4-2</td>
<td><table><tr><td>82</td><td>29</td><td>18</td><td>13</td><td>7</td></tr><tr><td>16</td><td>20</td><td>18</td><td>9</td><td>10</td></tr><tr><td>14</td><td>35</td><td>34</td><td>10</td><td>7</td></tr><tr><td>7</td><td>20</td><td>6</td><td>14</td><td>3</td></tr><tr><td>6</td><td>14</td><td>4</td><td>6</td><td>4</td></tr></table></td>
<td><table><tr><td>29</td><td>24</td><td>26</td><td>10</td><td>4</td></tr><tr><td>31</td><td>34</td><td>33</td><td>20</td><td>12</td></tr><tr><td>25</td><td>32</td><td>21</td><td>11</td><td>8</td></tr><tr><td>6</td><td>18</td><td>17</td><td>4</td><td>2</td></tr><tr><td>7</td><td>13</td><td>9</td><td>4</td><td>0</td></tr></table></td>
<td><table><tr><td>60</td><td>32</td><td>10</td><td>5</td><td>0</td></tr><tr><td>68</td><td>44</td><td>26</td><td>6</td><td>6</td></tr><tr><td>28</td><td>29</td><td>14</td><td>5</td><td>3</td></tr><tr><td>24</td><td>13</td><td>9</td><td>1</td><td>0</td></tr><tr><td>6</td><td>4</td><td>7</td><td>0</td><td>0</td></tr></table></td>
<td><table><tr><td>26</td><td>43</td><td>20</td><td>15</td><td>6</td></tr><tr><td>38</td><td>70</td><td>18</td><td>17</td><td>8</td></tr><tr><td>13</td><td>30</td><td>11</td><td>13</td><td>4</td></tr><tr><td>10</td><td>24</td><td>11</td><td>5</td><td>0</td></tr><tr><td>2</td><td>9</td><td>3</td><td>2</td><td>2</td></tr></table></td></tr>
<tr><td>4-2-2-2</td>
<td><table><tr><td>31</td><td>21</td><td>27</td><td>15</td><td>9</td></tr><tr><td>23</td><td>24</td><td>45</td><td>9</td><td>11</td></tr><tr><td>24</td><td>31</td><td>19</td><td>14</td><td>7</td></tr><tr><td>13</td><td>17</td><td>14</td><td>6</td><td>1</td></tr><tr><td>8</td><td>11</td><td>12</td><td>5</td><td>3</td></tr></table></td>
<td><table><tr><td>41</td><td>21</td><td>20</td><td>10</td><td>6</td></tr><tr><td>29</td><td>47</td><td>37</td><td>17</td><td>7</td></tr><tr><td>27</td><td>35</td><td>22</td><td>9</td><td>10</td></tr><tr><td>12</td><td>13</td><td>5</td><td>9</td><td>4</td></tr><tr><td>5</td><td>7</td><td>5</td><td>2</td><td>0</td></tr></table></td>
<td><table><tr><td>59</td><td>34</td><td>19</td><td>4</td><td>0</td></tr><tr><td>68</td><td>45</td><td>23</td><td>7</td><td>1</td></tr><tr><td>35</td><td>33</td><td>7</td><td>8</td><td>3</td></tr><tr><td>14</td><td>20</td><td>4</td><td>3</td><td>0</td></tr><tr><td>3</td><td>6</td><td>1</td><td>2</td><td>1</td></tr></table></td>
<td><table><tr><td>57</td><td>40</td><td>20</td><td>10</td><td>8</td></tr><tr><td>38</td><td>43</td><td>27</td><td>14</td><td>4</td></tr><tr><td>23</td><td>29</td><td>16</td><td>18</td><td>2</td></tr><tr><td>6</td><td>12</td><td>10</td><td>4</td><td>0</td></tr><tr><td>4</td><td>9</td><td>4</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>4-2-3-1</td>
<td><table><tr><td>74</td><td>55</td><td>40</td><td>11</td><td>2</td></tr><tr><td>33</td><td>39</td><td>31</td><td>16</td><td>6</td></tr><tr><td>28</td><td>18</td><td>16</td><td>10</td><td>3</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>400</td></tr></table></td>
<td><table><tr><td>68</td><td>58</td><td>40</td><td>11</td><td>6</td></tr><tr><td>49</td><td>42</td><td>26</td><td>9</td><td>7</td></tr><tr><td>15</td><td>16</td><td>17</td><td>7</td><td>2</td></tr><tr><td>4</td><td>6</td><td>5</td><td>4</td><td>2</td></tr><tr><td>0</td><td>2</td><td>3</td><td>0</td><td>0</td></tr></table></td>
<td><table><tr><td>158</td><td>64</td><td>18</td><td>4</td><td>1</td></tr><tr><td>62</td><td>37</td><td>12</td><td>2</td><td>1</td></tr><tr><td>19</td><td>4</td><td>5</td><td>0</td><td>0</td></tr><tr><td>2</td><td>6</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td>
<td><table><tr><td>88</td><td>74</td><td>34</td><td>11</td><td>4</td></tr><tr><td>42</td><td>46</td><td>29</td><td>15</td><td>3</td></tr><tr><td>12</td><td>16</td><td>6</td><td>5</td><td>4</td></tr><tr><td>1</td><td>3</td><td>3</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>4-3-3</td>
<td><table><tr><td>46</td><td>36</td><td>17</td><td>9</td><td>5</td></tr><tr><td>40</td><td>43</td><td>30</td><td>14</td><td>8</td></tr><tr><td>23</td><td>29</td><td>19</td><td>11</td><td>2</td></tr><tr><td>10</td><td>15</td><td>5</td><td>7</td><td>3</td></tr><tr><td>4</td><td>13</td><td>5</td><td>4</td><td>2</td></tr></table></td>
<td><table><tr><td>59</td><td>43</td><td>35</td><td>8</td><td>8</td></tr><tr><td>50</td><td>41</td><td>23</td><td>12</td><td>2</td></tr><tr><td>15</td><td>23</td><td>5</td><td>3</td><td>0</td></tr><tr><td>13</td><td>12</td><td>5</td><td>3</td><td>0</td></tr><tr><td>5</td><td>6</td><td>3</td><td>1</td><td>0</td></tr></table></td>
<td><table><tr><td>100</td><td>55</td><td>18</td><td>7</td><td>1</td></tr><tr><td>63</td><td>41</td><td>10</td><td>9</td><td>0</td></tr><tr><td>26</td><td>21</td><td>5</td><td>4</td><td>1</td></tr><tr><td>12</td><td>7</td><td>6</td><td>1</td><td>1</td></tr><tr><td>2</td><td>4</td><td>4</td><td>2</td><td>0</td></tr></table></td>
<td><table><tr><td>61</td><td>51</td><td>20</td><td>5</td><td>5</td></tr><tr><td>38</td><td>51</td><td>28</td><td>9</td><td>6</td></tr><tr><td>18</td><td>37</td><td>10</td><td>4</td><td>2</td></tr><tr><td>7</td><td>9</td><td>6</td><td>7</td><td>3</td></tr><tr><td>1</td><td>8</td><td>1</td><td>2</td><td>2</td></tr></table></td></tr>
</table>

Table 8: FANM-Debrecen Handsomes' crystal ball notation.

the statistical tests applied.

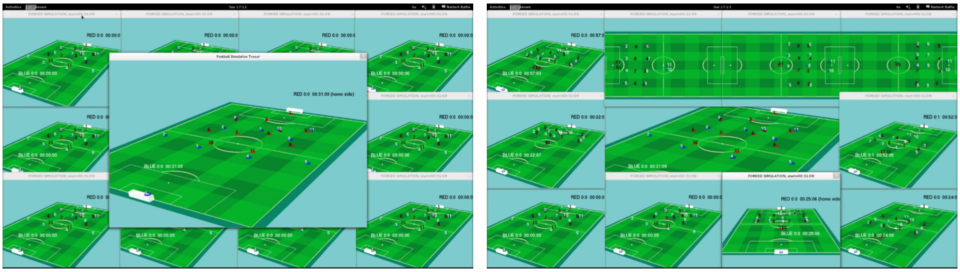
Season	1.	2.	3.	4.	5.	6.	7.	8.	9.	Test stat.
Goals	693	628	661	639	665	682	708	630	701	10/56
Points	668	655	659	659	665	658	666	660	669	11/38

Table 9: The total number of goals and points obtained from the simulation algorithm FANM-TF ($\overline{x}_g = 667.44$, $s^*_{n\,g} = 30.51$, $\overline{x}_p = 662.11$, and $s^*_{n\,p} = 4.96$). The last column shows the values of test statistics in form of Wald-Wolfowitz/Mann-Whitney.

It should be noted that all these simulations have been computed on the level "-1" in avatar mode.

7. Conclusion and Future Work

In this work, we introduced a general framework for sport science simulations that we refer to as Simulation Oriented Architecture (SimOA). As a concrete implementation of the framework, the basic soccer simulation algorithms investigated or used in the FootballAvatar project have been shown. The concept of avatars provides a solid basis for our simulations. It should be noted that more advanced simulation models have also been developed and implemented in our simulators. For example, fork-join soccer computations (see Fig. 7) or avatar clouds based cellular automata simulations will be presented in a further work.



(a) The time moment of starting the forked simulations.

(b) Further (heuristic) investigation of some selected passes.

Figure 7: All possible passes are simulated in different forked computations that can be seen around the large central soccer window.

8. Acknowledgments

The authors would like to thank Elemér Kondás, Sándor Szilágyi, Péter Szakály, and Tamás Sándor for contributing their professional expertise in football to this study. Sándor Szilágyi provided indispensable help with the foul model presented in Sect. 5.2. Similarly, the comments of Péter Szakály, Elemér Kondás, and Tamás Sándor regarding the biological and behavioural model (see Sect. 5.2) were essential. The authors also would like to thank Ferenc Frida and Géza Róka for their support and help. Last but not least, the authors are grateful to all members of the "Nagyerdei Gerundium" working group and other project partner companies (namely, Esantu Ltd., U1 Research Ltd., IQRS Ltd., and Satrax Ltd.) for the meetings and discussions about soccer.

NEW SZÉCHENYI PLAN

The publication was supported by the GOP-1.2.1-11-2012-0005 (*SziMe3D – 3D-s technológiai innovációk a turizmus, oktatás és sport területén*, *SziMe3D–3D technological innovation in tourism, education and sport*) project. The project has been supported by the European Union.

References

[1] N. Bátfai. Footballer and football simulation markup language and related simulation software development. *Journal of Computer Science and Control Systems*, 3(1):13–18, 2010.

[2] N. Bátfai. The socceral force. *CoRR*, abs/1004.2003, 2010. URL `http://arxiv.org/abs/1004.2003`.

[3] N. Bátfai, R. Dóczi, J. Komzsik, A. Mamenyák, Cs. Székelyhídi, J. Zákány, M. Ispány, and Gy. Terdik. Applications of a simplified protocol of RoboCup 2D soccer simulation. *Infocommunications Journal*, 5(1):15–20, 2013.

[4] N. Bátfai, P. Jeszenszky, A. Mamenyák, B. Halász, R. Besenczi, J. Komzsik, B. Kóti, G. Kövér, M. Smajda, C. Székelyhídi, T. Takács, G. Róka, and M. Ispány. Competitive Programming: a Case Study for Developing a Simulation-based Decision Support System. *Infocommunications Journal*, 8(1):24–38, 2016.

[5] K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change.* Addison-Wesley, second edition, 2004. ISBN 978-0-321-27865-4.

[6] M. Buckland. *Programming Game AI By Example.* Wordware Publishing Inc., 2005. ISBN 1556220782.

[7] J. L. Casti. *Would-Be Worlds: How Simulation Is Changing the Frontiers of Science*. John Wiley & Sons, 1997.

[8] A. C. Constantinou, N. E. Fenton, and M. Neil. pi-football: A Bayesian network model for forecasting association football match outcomes. *Knowledge-Based Systems*, 36:322–339, 2012. doi: 10.1016/j.knosys.2012.07.008.

[9] J. P. Davis, C. B. Bingham, and K. M. Eisenhardt. Developing theory through simulation methods. *The Academy of Management Review*, 32(2):480–499, 2007.

[10] M. Dixon and S. Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2):265–280, 1997. doi: 10.1111/1467-9876.00065.

[11] E. Epstein. A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, 8:985–987, 1969. doi: 10.1175/1520-0450(1969)008<0985:ASSFPF>2.0.CO;2.

[12] D. Forrest, J. Goddard, and R. Simmons. Odds-setters as forecasters: the case of english football. *International Journal of Forecasting*, 21(3):551–564, 2005. doi: 10.1016/j.ijforecast.2005.03.003.

[13] J. Goddard. Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, 21(2):331–340, 2005. doi: 10.1016/j.ijforecast.2004.08.002.

[14] J. Goddard and I. Asimakopoulos. Forecasting football results and the efficiency of fixed-odds betting. *Journal of Forecasting*, 23(1):51–66, 2004. doi: 10.1002/for.877.

[15] Google Inc. Protocol Buffers, 2014. URL `https://developers.google.com/protocol-buffers/`.

[16] A. Y. Govan and C. D. Meyer. Ranking national football league teams using Google's PageRank. Technical Report CRSC-TR06-19, Center for Research in Scientific Computation at North Carolina State University, 2006. URL `http://www.ncsu.edu/crsc/reports/ftp/pdf/crsc-tr06-19.pdf`.

[17] A. Y. Govan, C. D. Meyer, and R. Albright. Generalizing Google's PageRank to rank national football league teams. In *Proceedings of SAS Global Forum*, 2008. URL `http://meyer.math.ncsu.edu/Meyer/PS_Files/SASGF08RankingPaper.pdf`.

[18] D. Karlis and I. Ntzoufras. Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(3):381–393, 2003. doi: 10.1111/1467-9884.00366.

[19] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. RoboCup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, AGENTS'97, pages 340–347. ACM, 1997. doi: 10.1145/267658.267738.

[20] A. M. Law. *Simulation Modeling and Analysis.* Mcgraw-Hill, fifth edition, 2015.

[21] A. J. Lee. Modeling scores in the Premier League: is Manchester United really the best? *Chance*, 10(1):15–19, 1997.

[22] J. Martin. *Rapid Application Development.* Macmillan Publishing Co., 1991. ISBN 0-02-376775-8.

[23] I. McHale and P. Scarf. Modelling soccer matches using bivariate discrete distributions with general dependence structure. *Statistica Neerlandica*, 61 (4):432–445, 2007. doi: 10.1111/j.1467-9574.2007.00368.x.

[24] B. Mielczarek and J. Uzialko-Mydlikowska. Application of computer simulation modeling in the health care sector: A survey. *Simulation: Transactions of The Society for Modeling and Simulation International*, 88(2):197–216, 2012.

[25] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. URL `http://ilpubs.stanford.edu:8090/422/`.

[26] M. Peters, editor. *Computational Fluid Dynamics for Sport Simulation.* Lecture Notes in Computational Science and Engineering. Springer, 2009.

[27] RCSS Stamina Model, 1999. URL `http://people.dsv.su.se/~johank/RoboCup/manual/ver5.1release/browsable/browsablenode54.html`.

[28] A. Rotshtein, M. Posner, and A. Rakityanskaya. Football predictions based on a fuzzy model with genetic and neural tuning. *Cybernetics and Systems Analysis*, 41(4):619–630, 2005. doi: 10.1007/s10559-005-0098-4.

[29] H. Rue and O. Salvesen. Prediction and retrospective analysis of soccer matches in a league. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 49(3):399–418, 2000. doi: 10.1111/1467-9884.00243.

[30] J. Sewall, D. Wilkie, and M. C. Lin. Interactive hybrid simulation of large-scale traffic. *ACM Transactions on Graphics*, 30(6):135:1–135:12, 2011.

[31] P. Spagnoletti, S. Za, and R. Winter. Exploring foundations for using simulations in is research. In *Proceedings of the 34th International Conference on Information Systems (ICIS)*, 2013.

[32] SziMe3D Inc. FootballAvatar project, 2014. URL `https://footballavatar.hu`.

[33] The Australian Open Tennis Championship, 2014. URL `http://www.ausopen.com/index.html`.

[34] A. Tsakonas, G. Dounias, S. Shtovba, and V. Vivdyuk. Soft computing-based result prediction of football games. In *Proc. of the First International Conference on Inductive Modelling (ICIM'2002), Lviv*, pages 15–21, 2002.

[35] Union Cycliste Internationale. The Athlete Biological Passport, 2014. URL `http://www.uci.ch/clean-sport/the-athlete-biological-passport-abp/`.

[36] D. Wells. XP flow chart. URL `http://www.extremeprogramming.org/map/project.html`.