INTEGRA: An Open Tool To Support Graph-Based Change Pattern Analyses In Simulated Football Matches

Conference Paper · June 2020

DOI: 10.7148/2020-0228

Nicolò Oreste Pinciroli Vago
Politecnico di Milano

Yuri Lavinas
University of Tsukuba

Daniele Rodrigues
Pontifícia Universidade Católica de Campinas (PUC-Campinas)

Felipe Arruda Moura
Universidade Estadual de Londrina

INTEGRA: AN OPEN TOOL TO SUPPORT GRAPH-BASED CHANGE PATTERN ANALYSES IN SIMULATED FOOTBALL MATCHES

Nicolò Oreste Pinciroli Vago[1,2], Yuri Lavinas[3], Daniele Rodrigues[4], Felipe Moura[5], Sergio Cunha[6], Claus Aranha[3], Ricardo da Silva Torres[1]
[1]NTNU – Norwegian University of Science and Technology, Ålesund, Norway
[2]Dipartimento di Elettronica, Informazione e Bioingegneria – Politecnico di Milano, Milan, Italy
[3]University of Tsukuba, Tsukuba, Japan
[4]College of Computer Engineering, Pontifical Catholic University of Campinas, Brazil
[5]Laboratory of Applied Biomechanics, State University of Londrina, Londrina, Brazil
[6]College of Physical Education, University of Campinas, Campinas, Brazil

KEYWORDS

Simulated football matches, multi-agent systems, temporal graphs, graph visual rhythm.

ABSTRACT

This paper introduces Interactive Graph Analyzer (INTEGRA), a tool to support the comparison of simulated football matches with real ones, through the analysis of dynamic graphs. Our tool supports coordinated views of temporal graphs, benefiting from traditional node-link diagrams and graph visual rhythms, a recently proposed 2D image representation. Our proposal is generic and may be tailored to different applications. We demonstrate the use of this tool in compelling case studies related to the comparison of graph-based measurements obtained from three different kinds of simulated football matches and real ones. In particular, we exploit usage scenarios related to how graph measurements evolve over time.

INTRODUCTION

Multi-Agent (MA) system technologies have been applied successfully in several fields, such as games [Marín-Lora et al., 2020], robotics [Sharma et al., 2016], and medical research [Pathirana et al., 2019]. In the context of sport games, such technologies have been exploited to create realistic events and scenes aiming at improving users' experience. In particular, we highlight applications in players' interaction in football matches [Kitano et al., 1997], [Kurach et al., 2019], [Asada and von Stryk, 2020]

This paper targets the problem of creating more realistic MAs for football matches. The envisioned MA simulation is expected to allow the visualization, validation, and exploration of football player models, leading to a greater understanding of the relationship between the models and the real-world data, and extrapolations of many different scenarios using the rules derived from the models. The challenging problem is how to aggregate and represent different members of a team and their interactions [Machado et al., 2017] with the purpose of presenting to coaches and researchers a valuable tool to identify key-events and determinant moments of the matches, such as attacking sequences, shots to goal and tackles [Moura et al., 2012]. The resulting simulator is also expected to be used to help coaches and educators. The simulator will assist them with planning and decision making, by giving these professionals the tools to simulate fictional scenarios in the simulation and observe how these scenarios play out. The simulator will also allow them to obtain easily understandable outputs from the model in the form of videos of games between simulated agents, leading to an interactive process of trial, error, and discovery. Qualitatively and quantitatively assessing how different simulated football matches are from real-world ones is, therefore, of paramount importance.

This paper focuses on the comparison of simulated and real-world football matches based on tactical indicators defined in terms of position of players and their interaction (e.g., passes or proximity) on the pitch over time, a subject well studied in the Sport Science community [Duch et al., 2010], [Pena and Touchette, 2012], [Cho et al., 2018], [Mendes et al., 2018], [Oliveira and Clemente, 2018], [Buldú et al., 2018]. A widely used representation to support the identification and analysis of pattern changes associated with objects and their relationships (e.g., interaction of players in a football match) over time relies on the use of *temporal graphs* [Leskovec et al., 2005] and their proper visualization through visual representations [Beck et al., 2016].

In this paper, we propose Interactive Graph Analyzer (INTEGRA), a web-based visualization tool that supports coordinated views of dynamic graphs. This is a generic tool, which can be easily tailored to different applications. INTEGRA provides cus-

Communications of the ECMS, Volume 34, Issue 1,
Proceedings, ©ECMS Mike Steglich, Christian Mueller,
Gaby Neumann, Mathias Walther (Editors)
ISBN: 978-3-937436-68-5/978-3-937436-69-2(CD) ISSN 2522-2414

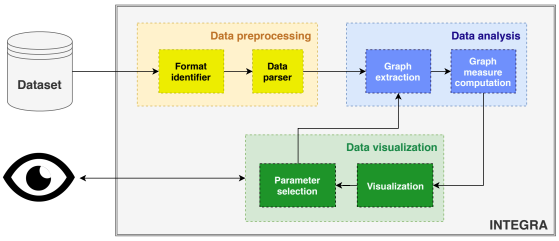


Fig. 1: Schematic architectural overview of INTEGRA.

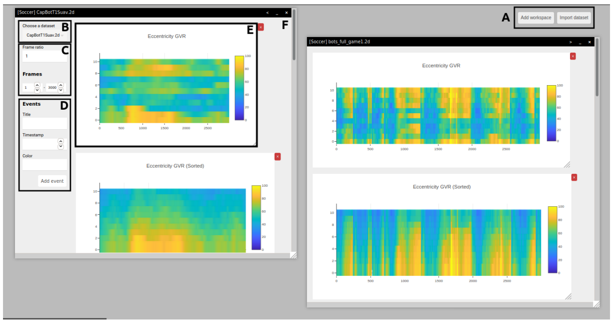


Fig. 2: Screenshot of the INTEGRA's interface.

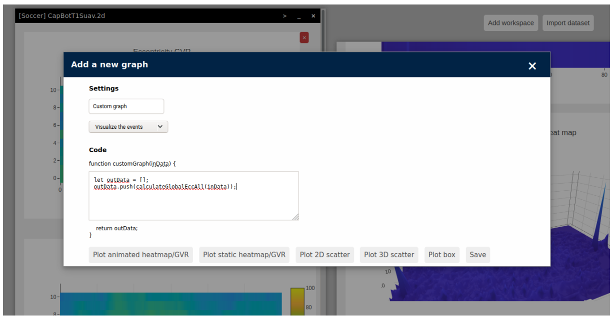


Fig. 3: Screenshot of the INTEGRA's interface for inserting temporal graphs manually.

tomizable interaction mechanisms and multiple visual representations based on Graph Visual Rhythms (GVRs) [Rodrigues et al., 2019] – a bidimensional representation which encodes graph features (e.g., vertex properties encoded through graph measurements) as columns of an image – and node-link diagrams.

Our approach focuses on allowing the user to customize the user interface, to interact with graphs, and to set custom parameters, on the basis of which analysis and visualization are performed. Using multiple views also expands the possibilities of identifying patterns of interest by coordinating different and complementary insights provided by different representations, obtained from both simulated and real-world matches. This paper describes the functionalities and architecture of the tool and illustrates its use in compelling usage scenarios, related to the comparison of data associated with three different kinds of simulated matches and real ones, based on graph measurements.

INTEGRA

INTEGRA architecture (shown in Figure 1) is divided into three modules: data pre-processing, data analysis, and data visualization.

Data pre-processing (in yellow) consists in the identification of the dataset format and in the subsequent parsing. In case of football games, the input data are converted into a matrix, that represents the players' coordinates over time. Data analysis (in blue), in turn, is in charge of extracting temporal graphs from the parsed dataset and of computing graph measurements. Later, extracted measurements will be used to compose visual structures in the data visualization module. A central feature of the tool consists in giving the user the possibility to write the code to plot custom graphs, using JavaScript. Data visualization (in green) allows the user to visualize the computed data and to customize the analysis and visualization parameters. The user may also interact with the graphs, for instance by selecting a specific range of values, by exporting the graphs as images and by playing animations. INTEGRA also allows creating several workspaces, that are represented as draggable and resizable window-like containers, so that users may compare the graphs of two or more datasets in the same browser window.

GUI Design

Figure 2 shows a screenshot of INTEGRA's interface. It shows that the GUI is divided into different workspaces. This workspace-based approach allows more sets of data to be studied at the same time, maintaining independent parameters for each workspace.

Components

Figures 2 and 3 show that the interface of the proposed tool is divided into the following sections:

- **Section A** contains the buttons that allow the user to import datasets in supported formats and to add new workspaces.
- **Section B** contains the menu that allows selecting one of the imported datasets for visualization. The modular architecture of INTEGRA allows the implementation of custom modules to parse different datasets, so that data can be properly handled using JavaScript. Currently, parsing is limited to the datasets for which specific parsers are included in the tool.
- **Section C** contains the menus to choose the frame ratio and the frame range. The frame ratio is the ratio between the current framerate and the original data framerate, while the frame range refers to the initial and final frames displayed in the visualization section.

• **Section D** is the *Events* section and it allows to add and delete custom events to the visualization. An event is defined as something that happens in correspondence of a specific timestamp. The events are shown in different ways on the basis of the graph type.
• **Section E** allows the customization of the visualization solutions and shows detailed information about the quantities represented in the graphs, for instance by zooming on a particular area or volume, by performing rotations and in some cases by starting animations.
• **Section F** shows the different visualization alternatives. The user may interact with the proposed graphs using the features available in Section E.
• The modal window *Add a new graph* (Figure 3) allows the user to create a customized graph on the basis of the imported data. In particular, it is possible to define and display the variables as colors, animations or axis variables. In order to create custom graphs, the user has to define the content of a JavaScript function.

Interaction
Initially, the user may create a new workspace or import one or several new files. Once the datasets have been imported, they can be selected from the left menu of a workspace. The result of the analysis will be shown on the graphs displayed next to the left menu.

It is also possible to create a custom graph by clicking on the + button in the graph area, once a specific dataset has been selected.

Implementation Aspects

The tool has been developed using HTML, CSS, and JavaScript in order to guarantee portability. Data analysis is performed using the local computational power, so that it is not necessary to rely on an Internet connection or on remote server availability.

The interface for a new workspace is dynamically generated using JavaScript. Firstly, a new workspace is generated on the basis of a default empty workspace. Then `EventListener`s are added, in order to allow the user to handle time and events.

In order to propose a modular structure, each workspace is described, at a given moment, by a set of status variables, that represent the values of the different parameters.

The proposed approach allows handling different kinds of datasets, since data access is independent of data elaboration and presentation. The tool supports several data formats and it is possible to implement and integrate other parsers in the code.

The tool is openly available at `https://github.com/nicolopinci/INTEGRA` (As of March 2020).

CASE STUDIES

In this section, we present two case studies about the use of INTEGRA for the analysis of simulated matches. In both cases, we use the proposed tool to perform comparative analyses among dataset associated with simulated matches (Google Research Football Environment) and a dataset related to a real football match.

Dataset Details

Simulated Match Dataset
The simulated match data were obtained from the Google Research Football environment, a recently published open-source football simulator [Kurach et al., 2019]. This simulator was originally proposed for the development of Artificial Intelligence Neural Networks. It reproduces a full football match with all of its usual regulations and events, as well as player tiredness, misses, etc. Figure 4 shows a standard image of the running simulator.

The entire simulated match lasts 5 real world minutes, and samples 10 frames from the environment per real world second, for a total of 3000 frames per game. Each frame has information about the position of each player, the position and possession of the ball, and player fatigue. This information is stored as a log file after the match. It is important to note that this simulator does not simulate data during interruptions in the game (fouls, offside, goals, and other referee-related interruptions). In these cases, the players are automatically placed on their positions when the game is re-started, and no information is recorded about the “off-play” period.

The simulator provides a standard, fixed strategy for controlling the agents (players), with three levels of strength (easy, medium, and hard). Additionally, the simulator allows a special controller (usually an AI controller) to take control of the player which is closest to the ball. For this case study, we produced games where all the players are controlled by the standard strategy (Bots Full), games where actions are selected at randomly disregarding any data from the simulator (Random Full) and games where the special controller is controlled by an Artificial Neural Network using the Proximal Policy Optimization algorithm (NN Full) [Schulman et al., 2017].

Figure 5 shows a visual output of the data acquired. The ball is shown as a red B, players from the same team are shown with the same letter, A and H. X is the player we control (in this case, a player of team H), while A is the team we play against.

Real Football Dataset
The considered dataset contains the position of the players for every timestamp. A total of 82,850 timestamps, related to 45 minutes (plus additional time) using a video framerate of 30 Hz, has been considered in the chosen dataset. Positional data were obtained using the DVideo software [Figueroa et al.,

Fig. 4: Screenshot of the football simulator used in this case study.

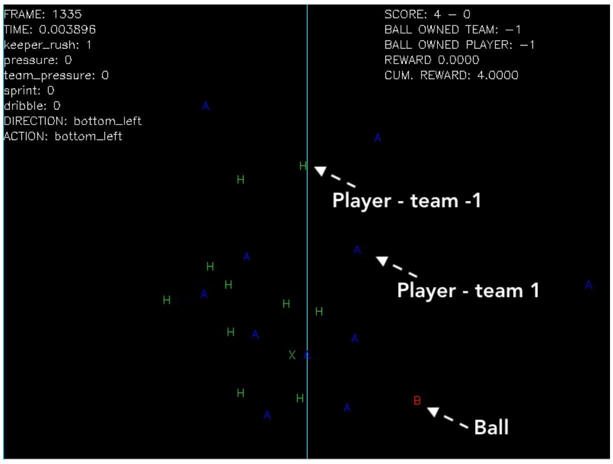


Fig. 5: Visual data from the simulator used by the Artificial Neural Network controller. The ball is shown as a red, the controlled player is shown as the green, team-mates are shown in green. Opponents are shown in blue.

2006b], [Figueroa et al., 2006a] applied to official football matches. In particular, the players' positions are computed for every timestamp by using a computer-vision-based tracking algorithm.

Tool Usage Overview

Figure 6 represents multiple views in a football match analysis scenario. Figure 6a represents the parameter selection. In particular, this section allows choosing a dataset, the frame ratio and the frame range. In the example, the user has selected a frame ratio of 1 (thus, the framerate is the same as the recording) and chose to visualize temporal graph information related to the period from frame 1 to frame 3000. Figure 6b presents the result of the parameter selection and some of the computed graphs. For each graph, a user can perform additional operations, such as selecting a specific area of interest. An example is shown in Figure 6c, where the GVR associated with the users' eccentricity scores is shown.

Graph Visual Rhythm

A Graph Visual Rhythm is a 2D representation that visually encodes temporal graph changes. This is a compact representation intended to highlight patterns of interest related to graph measurements in time.

Let $\mathcal{G} = \langle G_1, G_2, \ldots, G_T \rangle$ be a time evolving graph, where $G_t = (V_t, E_t)$ is an instant graph at timestamp $t \in [1, T]$ composed of a set of vertices, V_t, and a set of edges, E_t. A graph visual rhythm image GVR is defined as [Rodrigues et al., 2019]:

$$GVR(t, z) = \mathcal{F}(\mathcal{G}_t), \tag{1}$$

where $\mathcal{F}_{\mathcal{G}_t} : \mathcal{G} \to \mathbb{R}^n$ is a function that represents the instant graph $G_t \in \mathcal{G}$ as a point in an n-dimensional space, $t \in [1, T]$ and $z \in [1, n]$. $\mathcal{F}_{\mathcal{G}_t}$ can be defined as a graph measurement for each vertex of any other function that encodes relationships among vertices (e.g., degree histogram).

Eccentricity

Our case studies rely on the analysis of different graph patterns over time. In particular, we investigate the use of the eccentricity scores of players. In football matches, it measures the accessibility degree of a level from the other vertices, for a given time frame. Eccentricity, therefore, describes the spread degree of the players on the football field and how central a player is if compared to the other players of the same team.

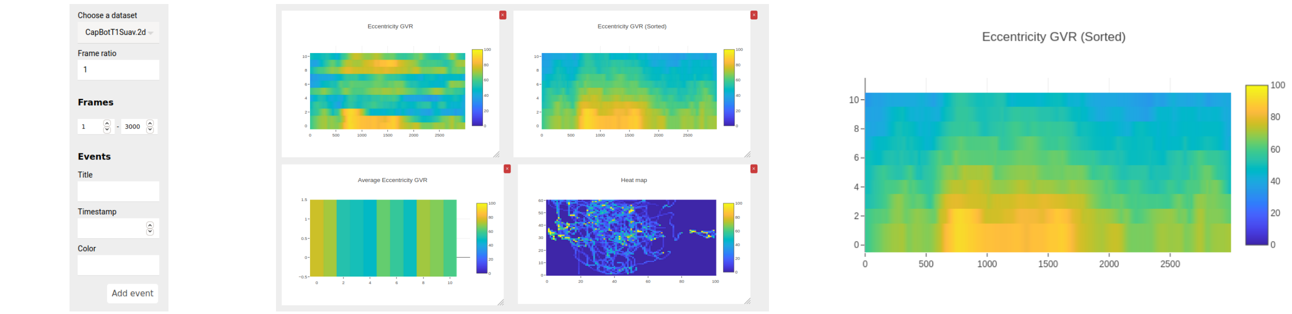
Given a graph $V(G)$, the eccentricity is a vertex measurement and corresponds to the maximum shortest distance from a vertex i to all others in the graph, as defined by [West, 2000].

$$\epsilon_i = max(d_{ij}) \tag{2}$$

where d_{ij} is the shortest distance between vertex i and j, where $j \in V(G)$.

On The Comparison Of Different Simulated Matches

Suppose that a user wants to compare the results of three different matches, which have been simulated using the Bots Full algorithm. Figure 7 shows the comparison between the eccentricity values of the three simulated matches based on the same Bots Full strategy, using GVR representations. For each GVR, we compute the eccentricity of players and sort their scores for each timestamp (GVR column). The GVR related to the dataset `bots_full_game1.2d` (top-left GVR) presents a large yellow area, which suggests that the players in that match are more connected to each other when compared to the players in the other two analyzed matches. The reason for this is that the football simulator, as real games, is a stochastic environment [Yue et al., 2008], causing the same action



(a) Parameter selection (b) Result of the parameter selection (c) Focus on a player's eccentricity

Fig. 6: Multiple views in football analysis scenario.

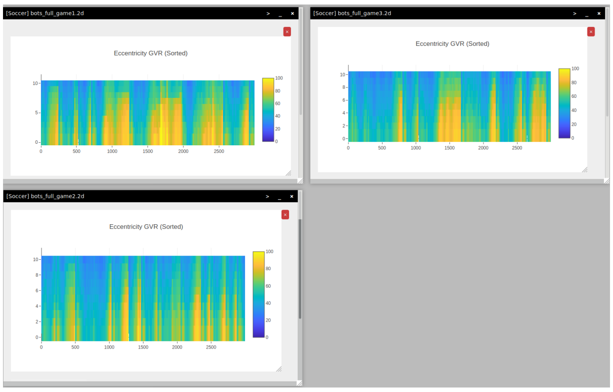


Fig. 7: Comparison between eccentricity GVRs of three Bots Full simulated matches.

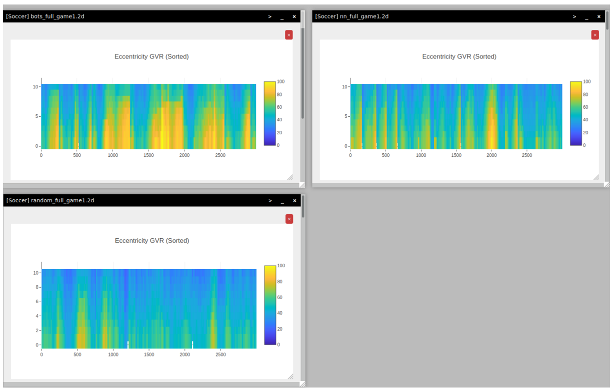


Fig. 8: Comparison between three eccentricity GVRs of three different simulated matches kinds: Bots Full (top-left), NN Full (top-right), and Random Full (bottom-left).

to have different outcomes at different times. Therefore, we can find different behaviors in matches that use the same strategy.

Suppose now that the user is interested in comparing different kinds of simulated matches. Figure 8 presents the comparison between the eccentricity values for a Bots Full simulated match (Bots Full, top-left GVR), a Neural Network simulated match (NN Full, top-right GVR) and a random agent (Random Full, bottom GVR). From those figures, we can observe that eccentricity scores for Bots Full are higher (more yellow areas) when compared to the others. The lack of yellow regions for the Random Full match may indicate that players are more disconnected in this kind of simulation. We may also observe a repetition pattern (yellow columns appear in a periodical frequency) for the NN Full match.

It comes with no surprise that eccentricity scores for Bots Full are high since the main goal of the bots Full player is to provide a realistic game play with reasonable football actions and strategies. On the other hand, the reason why the players in the match of the Random Full show low eccentricity scores might be because the actions of the controlled player are randomly selected, without any information about the environment, leading to a more sparse distribution of the players during the match. Finally, we comment about the repetition pattern of the NN Full match. This strategy was built with the only focus of scoring as many goals as possible, and the clear repetition pattern suggests that it found a combination of actions that, if repeated, allows the NN Full to score many goals.

On The Comparison Of Simulated And A Real football Match

Figure 9 presents the comparison between the real (bottom-right chart) and three simulated games using Bots Full (the other three plots) in terms of the eccentricity box plots. The main difference consists in the distribution of the eccentricity values: while the players' eccentricity presents several outliers (represented as points outside of the whiskers), the simulated games eccentricity values are almost entirely contained inside the whiskers' limits. The reason why this phenomenon is observed could be that the bots base their position on a common strategy, behaving like a swarm, while the players in real matches do not know or are not completely aware of the intentions of the other players in some of the defensive and offensive actions. In addition, it is important to emphasize that the real match dataset contains information about player positions even when the game is interrupted. Thus, it is possible that at

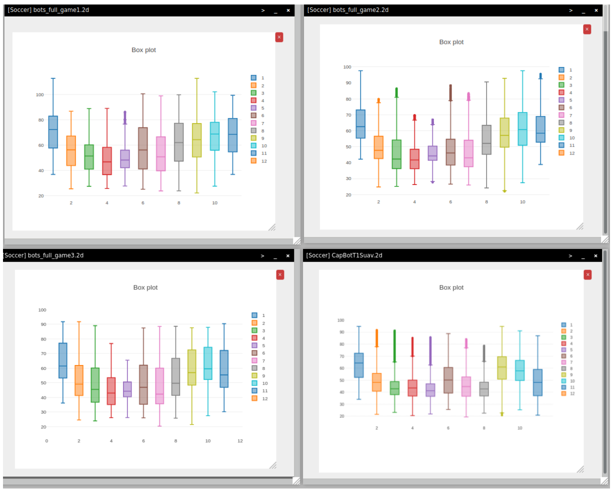


Fig. 9: Comparison between the box plots of completely simulated matches (top-left, top-right, and bottom-left) and of a real match (bottom-right). The real is associated with more outliers in the eccentricity values.

these moments, the values of eccentricity present discrepancies from the rest of the game. It is also possible to notice that the simulated game associated with the top-right plot is the most similar to the real one, with players from the same positions (i.e. goalkeepers and attackers) presenting similar behavior. Thus, this outcome suggests a more realistic simulation.

CONCLUSIONS

This paper introduced INTEGRA, a tool to support the analysis of simulated and real football match data based on visual analytics of temporal graphs. Its main novelty relies on permitting multiple views associated with the same dynamic graph, ranging from node-link diagram to graph visual rhythms. The tool is generic and can be tailored for different applications. It supports the implementation of various graph measures, as well as their visualization in 2D representations using graph visual rhythms.

The use of INTEGRA was illustrated in two scenarios related to the comparisons of graph-based dynamic evolution involving data of different simulated and real football matches. The tool supports the identification of similar and different patterns observed in football dynamics, defined in terms of player positions throughout the match and the behaviour of the players at prominent events in a match. This should allow coaches to have insights about successful and unsuccessful tactical strategies, possibly incorporating lessons learned in future decision-making processes.

Future work includes the evaluation of the tool with possible users. We also plan to investigate its use in different comparison scenarios, involving other tactical indicators proposed in the literature [Moura et al., 2012], [Moura et al., 2016].

ACKNOWLEDGEMENTS

Authors are grateful to CNPq and São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2015/24494-8, #2018/15178-3, #2016/50250-1, # 2018/19007-9, and #2017/20945-0). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[Asada and von Stryk, 2020] Asada, M. and von Stryk, O. (2020). Scientific and technological challenges in robocup. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):null.

[Beck et al., 2016] Beck, F., Burch, M., Diehl, S., and Weiskopf, D. (2016). A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*.

[Buldú et al., 2018] Buldú, J. M., Busquets, J., Martínez, J. H., Herrera-Diestra, J. L., Echegoyen, I., Galeano, J., and Luque, J. (2018). Using network science to analyse football passing networks: dynamics, space, time and the multilayer nature of the game. *Frontiers in psychology*, 9:1900.

[Cho et al., 2018] Cho, Y., Yoon, J., and Lee, S. (2018). Using social network analysis and gradient boosting to develop a soccer win–lose prediction model. *Engineering Applications of Artificial Intelligence*, 72:228 – 240.

[Duch et al., 2010] Duch, J., Waitzman, J. S., and Amaral, L. A. N. (2010). Quantifying the performance of individual players in a team activity. *PloS one*, 5(6):e10937.

[Figueroa et al., 2006a] Figueroa, P. J., Leite, N. J., and Barros, R. M. (2006a). Background recovering in outdoor image sequences: An example of soccer players segmentation. *Image and Vision Computing*, 24(4):363 – 374.

[Figueroa et al., 2006b] Figueroa, P. J., Leite, N. J., and Barros, R. M. (2006b). Tracking soccer players aiming their kinematical motion analysis. *Computer Vision and Image Understanding*, 101(2):122 – 135.

[Kitano et al., 1997] Kitano, H., Tambe, M., Stone, P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., and Asada, M. (1997). The robocup synthetic agent challenge 97. In *Robot Soccer World Cup*, pages 62–73. Springer.

[Kurach et al., 2019] Kurach, K., Raichuk, A., Stańczyk, et al. (2019). Google research football: A novel reinforcement learning environment. *arXiv preprint arXiv:1907.11180*.

[Leskovec et al., 2005] Leskovec, J., Kleinberg, J., and Faloutsos, C. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge discovery in data mining*, pages 177–187. ACM.

[Machado et al., 2017] Machado, V., Leite, R., Moura, F., Cunha, S., Sadlo, F., and Comba, J. L. (2017). Visual soccer match analysis using spatiotemporal positions of players. *Computers & Graphics*, 68:84 – 95.

[Marín-Lora et al., 2020] Marín-Lora, C., Chover, M., Sotoca, J. M., and García, L. A. (2020). A game engine to make games as multi-agent systems. *Advances in Engineering Software*, 140:102732.

[Mendes et al., 2018] Mendes, B., Clemente, F. M., and Maurício, N. (2018). Variance in prominence levels and in patterns of passing sequences in elite and youth soccer players: a network approach. *Journal of human kinetics*, 61(1):141–153.

[Moura et al., 2012] Moura, F. A., Martins, L. E. B., Anido, R. D. O., Barros, R. M. L. D., and Cunha, S. A. (2012). Quantitative analysis of brazilian football players' organisation on the pitch. *Sports Biomechanics*, 11(1):85–96. PMID: 22518947.

[Moura et al., 2016] Moura, F. A., van Emmerik, R. E. A., Santana, J. E., Martins, L. E. B., de Barros, R. M. L., and Cunha, S. A. (2016). Coordination analysis of players' distribution in football using cross-correlation and vector coding techniques. *Journal of Sports Sciences*, 34(24):2224–2232. PMID: 27079483.

[Oliveira and Clemente, 2018] Oliveira, P. and Clemente, F. M. (2018). Network properties and performance variables and their relationships with distance covered during elite soccer games. *Journal of Physical Education and Sport*, 18:1045–1049.

[Pathirana et al., 2019] Pathirana, S., Asirvatham, D., and Johar, M. G. M. (2019). Applicability of multi-agent systems for electroencephalographic data classification. *Procedia Computer Science*, 152:36 – 43. International Conference on Pervasive Computing Advances and Applications-PerCAA 2019.

[Pena and Touchette, 2012] Pena, J. L. and Touchette, H. (2012). A network theory analysis of football strategies. *arXiv preprint arXiv:1206.6904*.

[Rodrigues et al., 2019] Rodrigues, D. C. U. M., Moura, F. A., Cunha, S. A., and da Silva Torres, R. (2019). Graph visual rhythms in temporal network analyses. *Graphical Models*, 103.

[Schulman et al., 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[Sharma et al., 2016] Sharma, K. R., Honc, D., Dusek, F., and Kumar, T. G. (2016). Frontier based multi robot area exploration using prioritized routing. In Claus, T., Herrmann, F., Manitz, M., and Rose, O., editors, *30th European Conference on Modelling and Simulation, ECMS 2016, Regensburg, Germany, May 31 - June 3, 2016, Proceedings*, pages 25–30. European Council for Modeling and Simulation.

[West, 2000] West, D. B. (2000). *Introduction to Graph Theory*. Prentice Hall, 2 edition.

[Yue et al., 2008] Yue, Z., Broich, H., Seifriz, F., and Mester, J. (2008). Mathematical analysis of a soccer game. part i: Individual and collective behaviors. *Studies in Applied Mathematics*, 121(3):223–243.

NICOLÒ ORESTE PINCIROLI VAGO is a MSc Simulation and Visualization student at NTNU and Computer Science and Engineering student at Politecnico di Milano, where he obtained a BSc in the same field. His main research interests are genetic algorithms and Artificial Neural Networks applied to computer vision.

YURI LAVINAS is a PhD student at the University of Tsukuba. He graduated in Computer Science (2016) in the University of Brasilia, Brazil, and got his Master Degree at the University of Tsukuba (2020). His research focus on Multi Objective Evolutionary Computation, neuro-evolution and Artificial Life.

DANIELE RODRIGUES obtained her PhD in Computer Science from the University of Campinas (Unicamp) in 2017, Masters degree in Software Engineering from University of Campinas in 2004, and Bachelor Degree in Computer Engineering from Pontifical Catholic University of Campinas in 2001. She is currently a professor at College of Computer Engineering at Pontifical Catholic University of Campinas (PUC-Campinas). Her research focus on complex networks, machine learning, data science and sports science.

FELIPE MOURA is Professor at the Sports Sciences Department at State University of Londrina (Brazil). He graduated in Physical Education (2003) and got his Masters degree (2006) at São Paulo State University. He got his PhD in Physical Education at the University of Campinas (2011). Nowadays, he is the leader of the Laboratory of Applied Biomechanics in which the main research interests are biomechanics applied to sport and exercise, and signal processing of biological data.

SERGIO CUNHA has a degree in Physical Education, a master's degree in and a Ph'd in Sports Sciences from the State University of Campinas. He was a Visiting Professor at the "University of Calgary" for eight months, working on a research project together with Prof. Walter Herzog. He was also at the "University of Groningen" (Netherlands) as a Visiting Professor during 2019. He is currently an Associate Professor III at the School of Physical Education at Unicamp. He has experience in Physical Education and Sports, with an emphasis on Biomechanical Analysis Methods, working mainly on the following topics: biomechanics, football, futsal and mathematical models for sports.

CLAUS ARANHA obtained his PhD from the University of Tokyo in 2010, and is currently an assistant professor at the University of Tsukuba, and a member of the Center for Artificial Intelligence (C-AIR). His research focus on the study and application of Evolutionary Computation, including optimization, procedural generation, and artificial life simulations.

RICARDO DA SILVA TORRES is Professor in Visual Computing at the Norwegian University of Science and Technology (NTNU). He used to hold a position as a Professor at the University of Campinas, Brazil (2005 - 2019). Dr. Torres received a B.Sc. in Computer Engineering from the University of Campinas, Brazil, in 2000 and his Ph.D. degree in Computer Science at the same university in 2004. Dr. Torres has been developing multidisciplinary eScience research projects involving Multimedia Analysis, Multimedia Retrieval, Machine Learning, Databases, Information Visualisation, and Digital Libraries. Dr. Torres is author/co-author of more than 200 articles in refereed journals and conferences and serves as a PC member for several international and national conferences. Currently, he has been serving as Senior Associate Editor of the IEEE Signal Processing Letters and Associate Editor of the Pattern Recognition Letters.