

HELIOS2010 Team Description

Hidehisa Akiyama and Hiroki Shimora

Information Technology Research Institute, AIST, Japan
{hidehisa.akiyama,h.shimora}@aist.go.jp

Abstract. HELIOS2010 is a 2D soccer simulation team which has been participating in the RoboCup competition since 2000. This paper describes the overview and the current effort of HELIOS2010. Our current research focus is a game analysis and an online tactics planning. We are trying to apply unsupervised learning methods to extract game characteristic online. We show the result of preliminary experiment as a current effort. And, we also trying to construct a framework that enables agent to plan their tactics online.

1 Introduction

HELIOS2010 is a simulation soccer team for the RoboCup soccer 2D simulator(RCSS)¹[1]. We have been participating in the RoboCup competition since 2000 and reached 2nd place in RoboCup2009. We are developing several software that help us to develop a team and to make an experimental environment. Almost all of them have already been freely available as an open source software.

Our current research focus is a game analysis and an online tactics planning. In previous researches, we have proposed a novel agent positioning mechanism. Now, we are trying to utilize it to analyze unknown team formations. And, we are also trying to construct a framework that enables agent to plan their tactics online.

The setup of this paper is as follows. In section 2, we will introduce the overview of our released software. In section 3, we will describe our positioning mechanism and approach for the formation analysis. In section 4, we will describe our agent framework that enables online tactics planning. And, in section 5, we will end with a conclusion and future directions.

2 Released Software

We are developing several program packages as an open source software. Our software have high portability because all software are written in Standard C++, and support POSIX or use Qt². Now, we are mainly maintaining following three software packages:

¹ The RoboCup Soccer Simulator: <http://sserver.sourceforge.net/>

² A cross-platform application and UI framework: <http://qt.nokia.com/>

- librcsc: a base library for RCSS, which can be used as a framework for a simulated soccer team.
- soccerwindow2: a viewer program for RCSS. soccerwindow2 can work as a monitor client, a log player and a visual debugger. soccerwindow2 contains one more component, "fedit", that is an formation editing tool using human's intuitive operations.
- agent2d: a sample problem based on librcsc. agent2d can work as a simple simulated soccer team. Newbies can use agent2d as a good start point for developing their new teams.

These software are implemented from scratch without any source code of other simulated soccer teams. But, several idea were inspired from other teams' released code, such as CMUnited[2], YowAI[3, 4], TsinguAeolus[5], UvA Trilearn[6] and Brainstormers[7]. Thanks for their contributions.

These three software packages have already been freely available³. We hope that our released software help new teams to participate RoboCup events and to start a research of multiagent systems. But, please note that these software are licensed under LGPL or GPL. If you use them and publish the binary, you also have to release your source code.

3 Triangulation based Positioning Mechanism and Formation Analysis

We have proposed a function representation model to define a team formation[8, 9], which utilizes Delaunay Triangulation and a linear interpolation algorithm. The main idea of our model is similar to Situation Based Strategic Positioning[10], but our model achieved more flexibility. Now, we are trying to utilize our model to analyze unknown team formations.

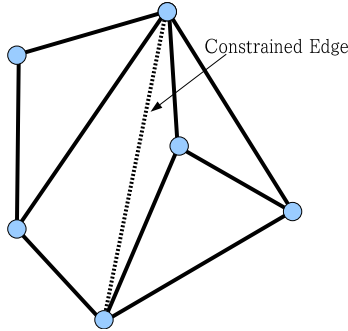


Fig. 1. Constrained Delaunay Triangulation.

³ RoboCup tools: <http://sourceforge.jp/projects/rctools/>



Fig. 2. The main window of fedit.

3.1 Positioning Mechanism

An input value of our positioning mechanism is a focal point on the soccer field, usually the ball position. And, output values are agents' strategic positions according to the input value. In our model, an input plane region (a soccer field) is divided into several triangles according to given training data. Such triangulated regions enable us to understand the influence of each training data easily.

Our latest implementation utilizes Constrained Delaunay Triangulation(CDT)[11] which is a generalized Delaunay Triangulation[12](figure 1). The difference between CDT and Delaunay Triangulation is the existence of constrained edges. If there are no constraints, the result of CDT is completely equivalent to the result of Delaunay Triangulation. As shown in the figure, unlike Delaunay Triangulation, CDT has some triangles that their circumcircle contains other triangles' vertices. CDT can restrict the topological relationship between two training data. This feature enables us to add a training data by not only a point but a line segment. As a result, we can compose more flexible team formation by using CDT.

3.2 Formation Editor

We have been developing a team formation editing tool, "fedit". fedit enables us to compose desired agents' positioning patterns according to the ball position by our intuitive operations. These data can be used as training data for our positioning mechanism. Figure 2 shows the screenshot of fedit. fedit can visualize the training process and enables us to edit the training data easily.

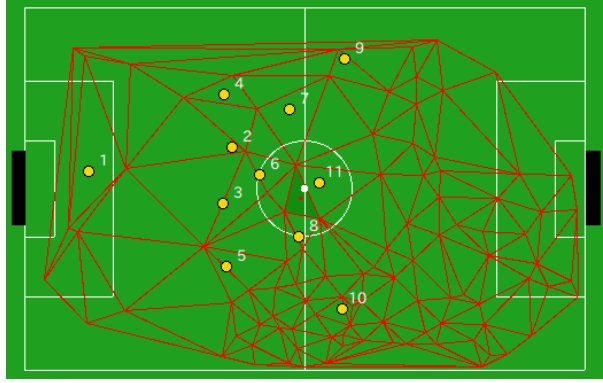


Fig. 3. An example result of formation analysis using GNG.

3.3 Formation Analysis using Unsupervised Learning Methods

One more advantage of using CDT is that we can easily utilize Neural Gas(NG)[13] or Growing Neural Gas(GNG)[14] for our model. These methods are known as an unsupervised learning method for vector quantization and can generate a set of (Constrained) Delaunay Triangulation that represents a characteristic of input point clouds. One of the advantages of these methods is its speed. We think these methods can be applied for online coach's game analysis, especially opponent team formation modeling, in realtime.

As a preliminary experiment, we applied NG and GNG in order to extract the characteristic of team formation from game log files. In this experiment, we used all `play_on` period in a game log file as input data. Input value for NG or GNG is 24 dimensional vector (coordinates value of ball and players). Finally, the result network is saved as our triangulation based formation model.

Figure 3 shows an example result of formation analysis using GNG(nearest unit step size: 0.5, connected unit step size: 10^{-4} , unit creation interval: 30, unit creation error decay: 0.5, error decay: 0.9, max train step: 12000, max unit size: 100). In this figure, the location of vertices are biased to the bottom-right area in the soccer field. Because NG and GNG can approximate a probability density function, this result means input ball locations were concentrated to the bottom-right area. This feature will be useful for online game analysis and dynamic adaptation.

Although we still need to improve these unsupervised learning methods and these methods have not been implemented in our online coach, we think these methods are promising to solve several problems of online formation analysis.

4 Action Chain Search Framework

We propose a framework to search a sequence of action using game tree search. In this framework, a sequence of action (such as direct pass, dribble or shoot)

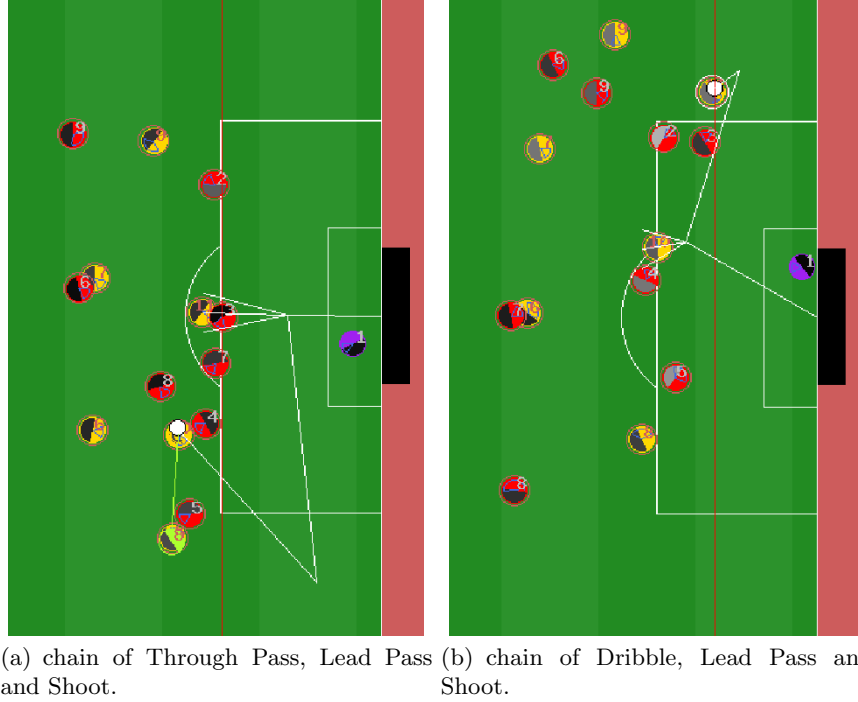


Fig. 4. Action Chain Examples.

is automatically searched on the fly. Figure 4 shows an example of sequence of action. Figure 4(a) shows a sequence consists of three actions, [through pass to teammate 8]–[lead pass to teammate 10]–[shoot]. Figure 4(b) shows a sequence consists of three actions, [dribble]–[lead pass to teammate 10]–[shoot].

With this framework, we need to prepare two kind of components, action generators and an action chain evaluator. An action generator generates a set of sequence of action-state pair. When an action is generated, the action generator also estimates the result state after the action. Usually, an action generator appends a candidate action to an existing sequence. Finally, an action generator returns sequences of action-state pair. For example, through-pass-action-generator searches possible through pass candidates. Action chain evaluator evaluates an action chain. Usually, an action chain evaluator evaluates the final state of action chain.

This framework has already been implemented in HELIOS2010. However, it has not performed effectively yet. The current problem is how to construct a good action chain evaluator. Now, we are trying to apply several machine learning method to obtain a good action chain evaluator.

5 Conclusion and Future Directions

This paper described the research focus and the current effort of HELIOS2010.

We have proposed a function representation model using triangulation based methods, in order to represent strategical multi-agent/robot positions. Now, we are trying to utilize our positioning mechanism to analyze unknown team formations based on unsupervised learning methods such as NG and GNG. We showed the preliminary experimental results. This work has not been used in our team yet, but we think our approach is promising for online game analysis.

We also trying to construct a framework that enables agent to plan their tactics online. The proposed framework has already been implemented, but has not worked effectively yet because it is difficult to obtain a good action chain evaluator. This is our important future work.

References

1. Noda, I., Matsubara, H.: Soccer server and researches on multi-agent systems. In Kitano, H., ed.: *Proceedings of IROS-96 Workshop on RoboCup*. (Nov. 1996) 1–7
2. Stone, P., Riley, P., Veloso, M.: The CMUnited-99 champion simulator team. In Veloso, M., Pagello, E., Kitano, H., eds.: *RoboCup-99: Robot Soccer World Cup III*. Volume 1856 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin (2000) 35–48
3. Suzuki, T.: Team YowAI Description. *RoboCup 99: Robot Soccer World Cup III* (2000)
4. Nakayama, K., Ako, T., Suzuki, T., Takeuchi, I.: Team YowAI-2002. *RoboCup 2002: Robot Soccer World Cup VI* (2002)
5. Yao, J., Chen, J., Cai, Y., Li, S.: Architecture of tsinghuaeolus. In: *RoboCup 2001: Robot Soccer World Cup V*, Springer-Verlag (2002) 491–494
6. UvA Trilearn 2003. <http://www.science.uva.nl/~jellekok/robocup/2003/>
7. Brainstormers 2d: Brainstormers Public Source Code Release. <http://www.ni.uos.de/index.php?id=1023>
8. Akiyama, H., Katagami, D., Nitta, K.: Training of agent positioning using humans’s instruction. Volume 11. (2007)
9. Akiyama, H., Noda, I.: Multi-agent positioning mechanism in the dynamic environment. In: *RoboCup 2007: Robot Soccer World Cup XI*. (2008) to appear.
10. Reis, L.P., Lau, N., Oliv  ira, E.: Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team. *Balancing Reactivity and Social Deliberation in MAS* (2001) 175–197
11. Chew, L.P.: Constrained delaunay triangulations. In: *SCG ’87: Proceedings of the third annual symposium on Computational geometry*, New York, NY, USA, ACM (1987) 215–222
12. Mark de Berg, Otfried Schwarzkopf, M.v.K., Overmars, M.: *Computational Geometry: Algorithms and Applications*. Second edn. Springer Verlag (2000)
13. Martinetz, T., Schulten, K.: A ”neural-gas” network learns topologies. *Artificial Neural Networks I* (1991) 397–402
14. Fritzke, B.: A growing neural gas network learns topologies. In: *Advances in Neural Information Processing Systems 7*, MIT Press (1995) 625–632