

# TokyoTechSFC 2006 Team Description

Hidehisa Akiyama, Hitoshi Hasegawa, Daisuke Katagami and Katsumi Nitta

Department of Computational Intelligence and Systems Science,  
Tokyo Institute of Technology  
Nagatsuta-cho, Midori-ku, Yokohama-shi, Kanagawa 226-8502, Japan  
{akiyama, hasegawa, katagami, nitta}@ntt.dis.titech.ac.jp  
WWW: <http://www.ntt.dis.titech.ac.jp/~akiyama/robocup/>

**Abstract.** This paper describes the research focus of TokyoTechSFC that is a 2D soccer simulation team and has participated in the RoboCup competition since 2000. We are trying to introduce a new positioning mechanism to realize a more realistic soccer team behavior.

## 1 Introduction

TokyoTechSFC is a 2D soccer simulation team that has been participating in the RoboCup competition since 2000. Our code is written by C++ and implemented from scratch without the source code of any other simulated soccer teams.

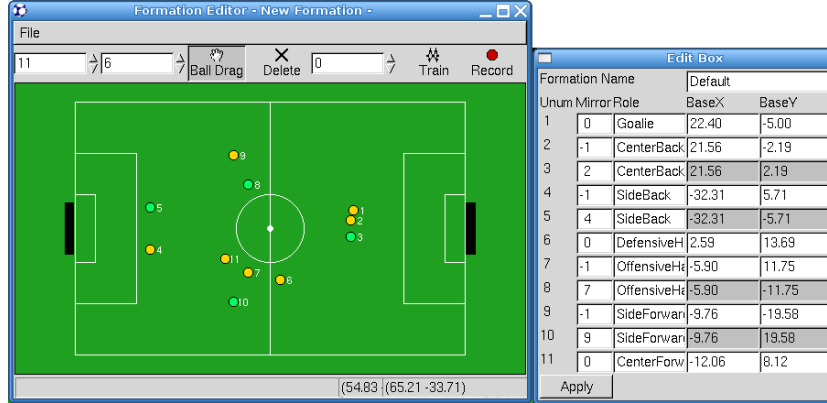
We are developing three program packages: 1) librcsc, the basic library for a simulated soccer agent and related tools, 2) TokyoTechSFC, the main agent programs that works as a simulated soccer team and 3) soccerwindow, the viewer program that works as a visual debugger and a human-agent interaction tool. The preliminary version of librcsc and soccerwindow have already been released<sup>1</sup>. We are planning to release the latest sources and maintain them. We hope that these help a new team to participate the RoboCup event and to start a research about the multi-agent simulation using the RoboCup soccer simulator.

Our main research focus is to realize a more realistic adaptation mechanism using human interactive coaching system. Currently, we are trying to improve a positioning mechanism concerned with not only a team formation but also a dynamic positioning. In the following sections, we describe the outline of our positioning system.

## 2 Formation Construction

We assume that a team formation is constructed by human trainer's instruction. We developed a GUI tool that helps us to construct a team formation. This tool, Formation Editor was implemented as one module of our viewer soccerwindow(Fig.1). Formation Editor has following features:

<sup>1</sup> <http://rctools.sourceforge.jp/>



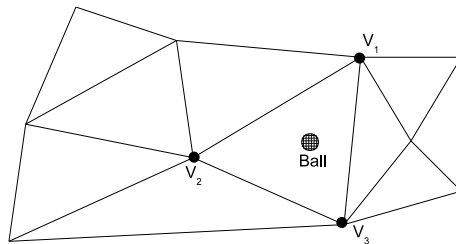
**Fig. 1.** Formation Editor

- It enables us to move the ball and players anywhere.  
We can move all objects using mouse. And we can directly specify the coordinate of object positions using the dialog window. Those position status can be stored as the training data.
- It can update the players position automatically.  
When we move the ball, all players will be moved based on the current ball position. This feature helps us to confirm the formation status.
- It can construct a team formation using the stored training data.

The basic idea of player agents' positioning rule in our methods is same as the Situation Based Strategic Positioning(SBSP)[2]. Input is the ball position and output is the position where player agents should move. In SBSP, player agents' positioning rule is defined by linear function and several parameters. On the other hand, we use nonlinear function to compute the positions of player agents. This feature enables us to make more complicated positioning rule for a team formation. We are testing some function approximators and/or interpolation methods with computational geometry algorithm to construct a formation from the stored training data.

At first, we applied a traditional three layer back-propagation neural network as a function approximator. Neural network worked well though it was a very simple implementation. Some complex formations like overcrowdedness to the penalty area was able to be achieved without any special rules. Moreover, if we use Formation Editor, this method hides not only the positioning rule but also the positioning parameters from the team developer. Therefore, it is not necessary for them to understand the team formation mechanism. In our experience, almost all testee that has no knowledge about the soccer simulation could construct a suitable team formation easily and quickly.

Neural network showed a good performance. However, it was difficult to adjust it in detail since the neural network causes an over-fitting. This is because all training data affect the result approximated function. So, it is preferable that



**Fig. 2.** Delaunay Triangulation

each training data is mutually independent. Now, we are trying to apply the new method to solve this problem. The interpolation method using computational geometry analysis will be a good solution. We implemented the Delaunay triangulation algorithm that computational complexity is  $O(n \log n)$  [1]. In our new method, all training data is used as the vertex point of the Delaunay triangle and the field is divided to several triangle mesh. If ball is in a triangle, the move target position of the player agent is computed as the linear interpolation result of three vertex(Fig.2). If ball is out of triangles, the nearest vertex will be used. This method is still testing, but will show a better performance than our old method or any other hand-made formations.

The triangulation will also show the interesting analytical result. We plan to use this technique for a game analysis and other coaching systems in our research.

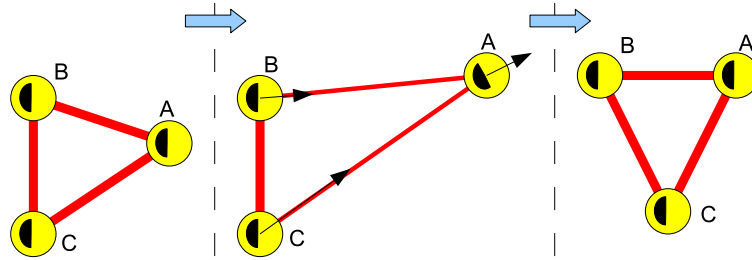
### 3 Dynamic Positioning

To realize a more realistic simulated soccer team, a positioning technique to balance a team formation dynamically is a important feature[2]. This dynamic positioning requires a positioning mechanism different from a team formation system. A key point is how to balance a distance between player agents. We propose the spring model to solve this problem.

Fig.3 shows a typical situation that a dynamic positioning is required and how the springs work. In fig.3, three players are connected with springs(shown by line with thickness). When player A goes to some direction, player B and C are pulled with the spring connected with player A. In this example, player C is pulled by stronger power because the spring has expanded more. Player C moves a distance that is longer than player B, and finally the distance between player agents is balanced.

In our model, each spring has its own strength, but the spring doesn't oscillate and gives only the vector that player agent should go. Now, we have three problems to realize our spring model.

First, we have to consider how to determine the players connection. The Delaunay triangulation technique described in the previous section may also become a good solution.



**Fig. 3.** Dynamic Positioning using Spring Model

Second, we have to consider how to determine the strength of each spring. We plan to apply some learning method to this problem.

Third, a dynamic positioning often requires the role exchange because the player agents position is very flexible and fluid. We should consider to update the connection when the player agents' geometric relation is different from the pre-defined connection topology. However, because player's sight is very limited, it is difficult to determine the update of the connection only by the individual player's decision. We will have to introduce a communication protocol to share the connection status.

## 4 Conclusion and Future Works

We are developing the training tool that enables human to design a team formation. This tool enables the beginner that has no knowledge about the soccer simulation to construct a suitable team formation easily and quickly. Now, we are implementing the interpolation method using the Delaunay triangulation to achieve a more accurate formation from the training data. We should compare the performance of our method with several function approximators and/or interpolation methods.

We are also trying to realize a dynamic positioning system. Because this project just has started, we have many problems to be solved. But, this will become the key technique in the future soccer simulation.

## References

1. Marc van Kreveld Mark de Berg, Otfried Schwarzkopf and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer Verlag, second edition, 2000.
2. Luis Paulo Reis, Nuno Lau, and Eugenio Oliv3ira. Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team. *Balancing Reactivity and Social Deliberation in MAS*, pages 175–197, 2001.