Identification and assignment of player roles in soccer

Kizuki Hasunuma s1240076 Supervised by Maxim Mozgovoy

Abstract

The present paper tries to further enhance soccer game artificial intelligence by adding information about the roles of forwards, defenders, goal keepers, etc. to soccer games artificial intelligence.

If the artificial intelligence does not have role information, it will be difficult to use even if save the movement of the artificial intelligence because it is difficult to organize which artificial intelligence plays which role. Therefore, it is expected that the use of artificial intelligence with role information will increase the number of situations where it can be used.

For this purpose, it is necessary to correctly assign the optimal role to each player. The role is automatically assigned by counting the number of roles played by the player and assigning the role with the largest number of roles. Also, verify that the assigned role is correct by comparing it to some manually configured roles.

1.Introduction

Sports Game artificial intelligence is used for various tasks such as sports analysis. This makes it easier to study the movements of sports players. I like sports and games very much and often play sports games. So I was interested in this research.

Artificial intelligence basically works as follows. If the soccer artificial intelligence has the ball, chose dribbling, passing or shooting. If artificial intelligence are near the ball, artificial intelligence will be defended if artificial intelligence are defensive, and if artificial intelligence are attacking you will receive a pass. All these actions are stored as data. In addition to these data, : each player is characterized with the following attributes: team ID, unique player ID, player jersey number, field coordinates, and moving speed.[1] Also, according to Large-Scale Analysis of Formations in Soccer[2], the use of "role-representation" will allow the movement of artificial intelligence to be accurate. However, these artificial intelligence do not have function to save automatically role information such as defenders, forward, goalkeepers and also. Because in the real soccer : There are 22 free-moving players. So we have to follow all the players at the same time to analyze the situation,a large number of calculations are needed. Consequently, the analysis of soccer games is not as developed as other sports , in part due to the overall complexity of soccer.[1]

Role information is essential for both actual sports games and game AI. These goalkeeper, forward and defender position data are very important.

2.Method

2.1.Assign role

2.1.1.Define role and Assign role number

To assign roles, to use the data that drove soccer game artificial intelligence for about 100 frames. Players are displayed as shown in Figure 1.

First, each time the frame is updated, assign a role to each player. The player in red is the attacker, and the player in blue is the defender. When assigning a role to a red players, the player with a higher vertical position on the right is assigned a smaller role number. Also,When assigning a role to a blue players, the player with a higher vertical position on the left is assigned a smaller role number. The role numbers are as shown in table1. Also, if assign roles to Figure 1 according to these rules, the role numbers are as shown in table 2.

Next, player number and role number assigned to that player are output as a file. The file format is output in the same format as Figure 2. The number on the left is the role number. And the number on the right is the player number. Figure 2 shows the player numbers created from Figure 1 and the role numbers assigned to the players. The following processing is performed for to assign a role. The reason from is 0 to 4 is that there are 4 defenders, 4 midfielders and 2 forwards.

```
 for (int from = 0, until = 4; from < 5; from += 4, until += 4)
  {
  std::sort(players.begin() + from, players.begin() +
        until, [teamID](game_object & p1, game_object & p2)
      {
  return teamID ? p1.get_position_y() < p2.get_position_y() :
                    p1.get_position_y() > p2.get_position_y();
      });
  }

std::sort(players.begin() + 8, players.end(),
            [teamID](game_object & p1, game_object & p2)
{
return teamID ? p1.get_position_y() < p2.get_position_y() :
                         p1.get_position_y() > p2.get_position_y()
});


int simulatorID = 3 + teamID; // start from left defender
for (size_t i = 0; i < players.size(); i++)
{
game_object player = players[i];
 players_ids.insert(std::pair<int,
                  int>(player.get_jersey_number(), simulatorID));
simulatorID += 2;
}
 return players_ids;
```

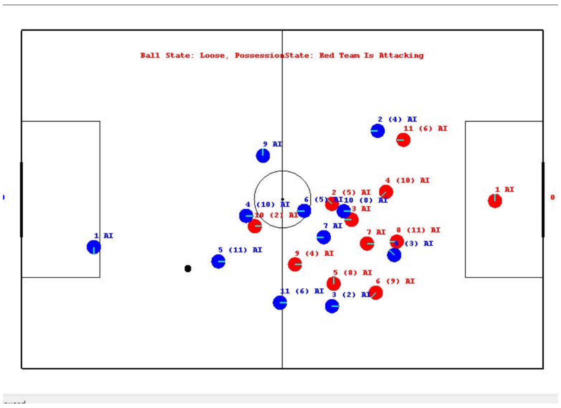


Figure1:Player mapping

Role Name	Role Number
Goal keeper	1
Left defender	2
Left center defender	3
Right center defender	4
Right defender	5
Left midfielder	6
Left center midfielder	7
Right center midfielder	8
Right midfielder	9
Left forward	10
Right forward	11

Table 1: Role name and role number

Red Team		Blue Team	
Player	Role	Player	Role
1	1	1	1
2	11	2	9
3	4	3	4
4	8	4	5
5	6	5	11
6	2	6	6
7	3	7	10
8	7	8	7
9	5	9	3
10	10	10	2

11	9	11	8

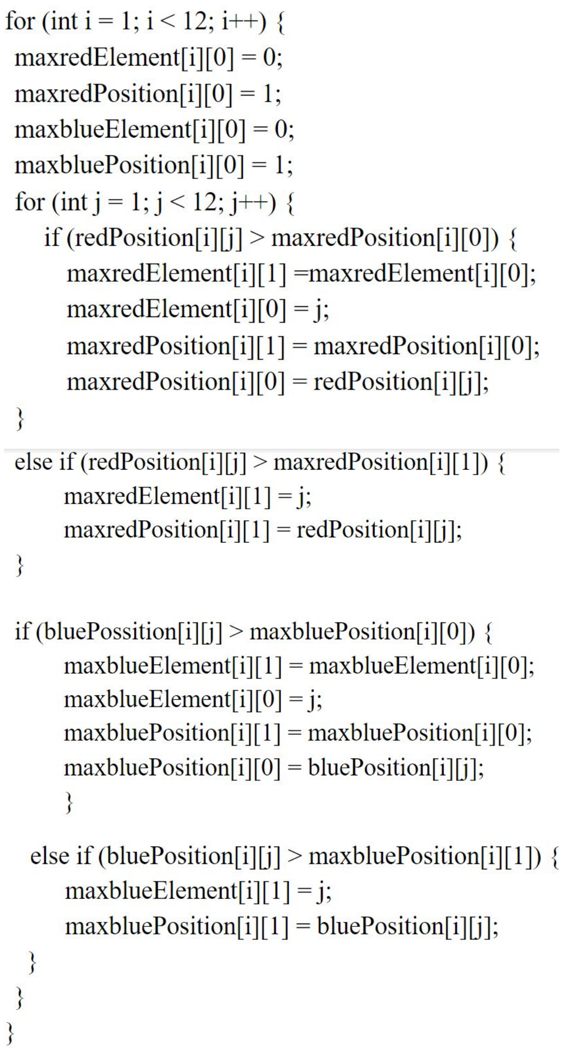
Table 2:Result of assigning roles to the players in Figure 1

2.1.2.Count role number and assign befitting role

Use all the player numbers and role numbers.

1) Count which role each player played and how many times. For each player, save which role performed and how many times, from the largest one.

2) The role that has been performed most frequently among the roles counted for each player is assigned to that player. The role with the highest number out of the counted numbers will be the role of that player. If there is more than one player who has the assigned role and has the same role, the role will be assigned to the player who has performed the role more. Unassigned players will be assigned the second most played role. For example, suppose left forwards were assigned to player numbers 2 and 5. If player number 2 is assigned to left forward 5 times and 2 times right forward, and player number 5 is assigned to 4 times left forward and 3 times to left defender, then left forward is assigned to player number 2. Also, left defender is assigned player number 5. Perform the following steps to compare the number of roles and assign the correct role.



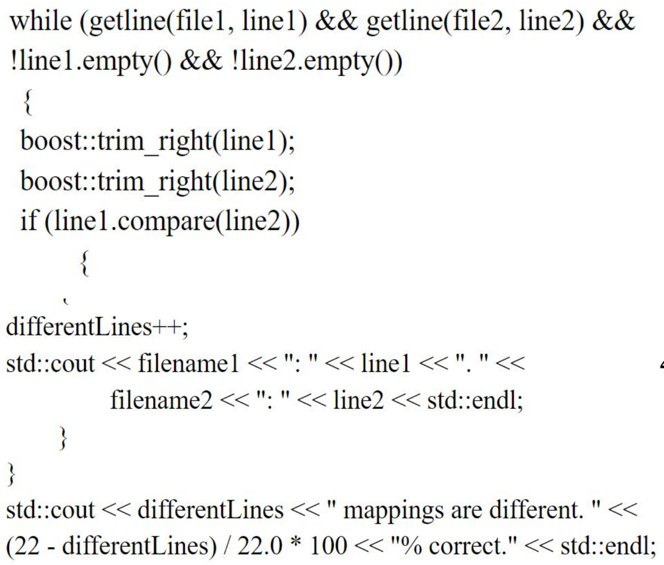
```
for (int i = 1; i < 12; i++) {
 maxredElement[i][0] = 0;
 maxredPosition[i][0] = 1;
 maxblueElement[i][0] = 0;
 maxbluePosition[i][0] = 1;
 for (int j = 1; j < 12; j++) {
    if (redPosition[i][j] > maxredPosition[i][0]) {
      maxredElement[i][1] =maxredElement[i][0];
      maxredElement[i][0] = j;
      maxredPosition[i][1] = maxredPosition[i][0];
      maxredPosition[i][0] = redPosition[i][j];
 }
 else if (redPosition[i][j] > maxredPosition[i][1]) {
      maxredElement[i][1] = j;
      maxredPosition[i][1] = redPosition[i][j];
 }

 if (bluePossition[i][j] > maxbluePosition[i][0]) {
      maxblueElement[i][1] = maxblueElement[i][0];
      maxblueElement[i][0] = j;
      maxbluePosition[i][1] = maxbluePosition[i][0];
      maxbluePosition[i][0] = bluePosition[i][j];
      }

  else if (bluePosition[i][j] > maxbluePosition[i][1]) {
      maxblueElement[i][1] = j;
      maxbluePosition[i][1] = bluePosition[i][j];
  }
 }
}
```

2.2.Check correct role

Used the file to manually assign roles to each player to make sure that the correct roles are assigned by comparing them to the roles that were automatically assigned to the players in 2.1.1. After comparison, prints out what the line did not match and percentage of the line did match. The result is displayed as shown below.

```
seq_0063_tm.txt:  4-8.  seq_0063_tm_auto.txt  4-4
seq_0063_tm.txt:  5-4.  seq_0063_tm_auto.txt  5-8
seq_0063_tm.txt:  7-9.  seq_0063_tm_auto.txt  7-10
seq_0063_tm.txt:  8-10.  seq_0063_tm_auto.txt  8-9
seq_0063_tm.txt:  6-6.  seq_0063_tm_auto.txt  6-5
seq_0063_tm.txt:  10-5. seq_0063_tm_auto.txt  10-6
6 mappings are different. 72.7273% correct.
```

Lines 1 to 6 show the numbers of roles that did not match the names of the compare files. It shows how much the last line matched.
The following processing is performed for the comparison.



```
while (getline(file1, line1) && getline(file2, line2) &&
!line1.empty() && !line2.empty())
 {
 boost::trim_right(line1);
 boost::trim_right(line2);
 if (line1.compare(line2))
      {
differentLines++;
std::cout << filename1 << ": " << line1 << ". " <<
          filename2 << ": " << line2 << std::endl;
      }
}
std::cout << differentLines << " mappings are different. " <<
(22 - differentLines) / 22.0 * 100 << "% correct." << std::endl;
```

3.Results

3.1.Assign collect role

100 dates created from different movements of soccer games artificial intelligence was used. Counted the number of times we performed roles after assigning roles for each frame. As a result of executing with 100 files, the number of roles was correctly counted and output in the correct format file.

3.2.Matched average of manually file and automatically files.

Use method 2.2. Compared 100 manually created files with 100 automatically created files. Each file is made from a soccer game of about 50 to 250 frames in length per game.
Printed out what the line did not match in the correct format and percentage of the line did match in the correct format. Also, calculated the average of how well the manually created files matched the automatically created files.
Result of Calculations show that the files created manually and the files created automatically have a 77.14% match on average.
This result is not good. Because about 23% of the roles are not properly assigned. 23% say about 4 to 5 players out of 22 players on the field are assigned the wrong role. This rarely happens when it is a real soccer game. So it's not good. These are that the assigned roles that each player has saved are only the top four of the roles that that player has been assigned, and it seems to be due to bad processing when all roles overlap with other players.

4.Conclusion

In this paper, considered the role information was added to the soccer game artificial intelligence, and the extent to which the role information was correctly given.we was able to give role information to soccer games artificial intelligence. Also, the result is that 77.14% of file created manually and file created automatically are in agreement. But it cannot be said that the information is completely correct. Since the number of saved roles is for the top three, it is possible that the correct role may not be assigned when there are players who will be repeatedly affected. Therefore, storing more roll count information and increasing the match between manually created files and automatically created files may improve accuracy.

5.References

[1]A. Moriyama, M. Mozgovoy. Classification and Clustering in Soccer Analytics. *Proceedings of the 47th International Conference on Control Processes and Stability*, St. Petersburg, Russia, 2016, vol. 3(19), pp. 576-582.

[2]Wei, X., Sha, L., Lucey, P., Morgan, S., & Sridharan, S. (2013, November). Large-scale analysis of formations in soccer. In *2013 international conference on digital image computing: techniques and applications (DICTA)* (pp. 1-8). IEEE.