Pass recognition in soccer game data

Takahiro Nakajima s1240193 Supervised by Prof. M. Mozgovoy

Abstract

Technology such as AI is being used for strategic analysis in sports. The purpose of this research is to make up a game engine that can simulate the game based on actual soccer game data. A system that simulates the movement of players has a function that recognizes that the action is a pass when the player receives the ball and updates the player who currently has the ball. If it do not recognize the ball correctly, the player holding the ball will make an error and we will not be able to formulate the correct team strategy. The problem is that sometimes it is judged as pass even though it is not a pass. For example, when the ball passes the player's head, the simulator determines a pass when the player and the ball overlap, even though the player is not actually touching the ball. I have tried two approaches to solve the problem. The first approach is to make a pass decision based on whether the player has held the ball for 3 frames or more after receiving the ball. This means that it takes 1 or 2 frames for the ball to pass over the player's head, so if the player holds the ball for more than 3 frames after receiving it, it can be considered a pass. The second approach is to make a pass judgment based on a change in the direction of the ball. If it is not a pass, the ball will pass through the player and the direction of the ball will not change. Therefore, from the state where the position of the player and the ball overlap on the simulator, if the direction of the ball changes, it is regarded as pass, and if not, it is determined that it is not pass. In the real soccer game data used, the players made 53 passes. In the actual soccer match used, the player made 582 passes. The number of passes determined in the simulation using the first method was 569, and the number of passes was 541 in the second method. In addition, the wrong pass was determined 13 times in the first method and 41 times in the second method. Therefore, the second approach, which uses the change of direction of the ball, turns out to be closer to the actual match path flow. Use this method for simulation to help analyze your soccer strategy.

1 Introduction

In Japan, the Tokyo Olympics will be held in 2020, and people's interest in sports has increased. In recent years, AI and other technologies have been used in sports from the perspective of analysis for team strategy and competition judgment.

Much of the actual soccer game information is converted into a datasets and saved for use in team strategies. The datasets contains information on the position of each player and the ball, as well as information on the number and ID of each player. Based on that information, we analyze player behavior and make pass judgments. The soccer game engine I'm researching can simulate the behavior of players during a soccer game by dividing them into 1 frame at a time based on the actual soccer game datasets [1]. That is, it can display information about which player has the ball for each frame and whether the player has passed the ball. The simulator displays the actions of the players during the soccer game as text. This is very efficient.

However, makeup is required for the pass judgment of this soccer game engine. The main problem is that if a non-pass action is judged as a pass action, the simulation result will be different from the actual game. For example, there is a problem that when a pass is made that flies over the player's head, the pass is recognized even though the player has not actually touched the ball.

The main cause of this problem is that the ball coordinates are stored in datasets in 2D instead of 3D. Since there is no material to determine the height of the ball from the ground, the simulator also determines that the ball passes higher than the player. Unless these Pass judgments are properly recognized, accurate team strategies cannot be performed. The goal of my research is to correct wrong pass decisions so that the algorithm proposed in [2] will result in the player's actual behavior and simulation results being the same. Doing this research will allow the game engine to simulate more realistic soccer games.

2 Method

2.1 Subjects

I used an application called TSG Simulator to simulate a soccer game (Figure 1).

It can display the Players' movements during the game one frame at a time by inputting the file where the data containing the position of the ball and position of the player in the game, or the number and ID of the player is stored. The team is divided into red and blue, and each player has a uniform number and ID. This player ID is used to identify each player. This simulator can simulate the movement of players in a game at a frame rate of 10 frames per second. The game data used for research consisted of 1479 frames in total, with only the red team attacking.

The problematic pass that described in the introduction is the so-called through pass, where the ball jumps over the player's head (Figure 2). When a pass action occurs, such as a ball passing through a player, there is a teammate between the player giving the pass and the player receiving the pass. In this case, the teammate will not hold the ball and will not be considered a pass. However, since this teammate is on the straight line where the pass is being made, the simulator will recognize it as a pass. The main problem that such a flow pass leads to wrong pass judgment.

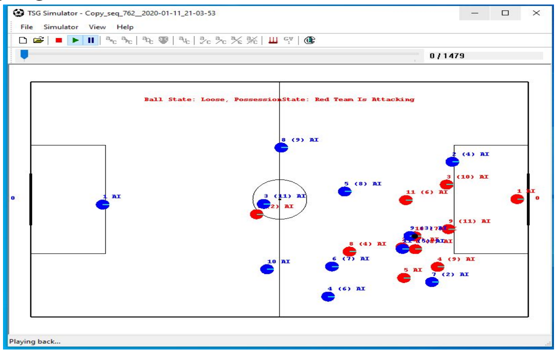
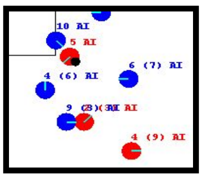
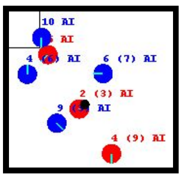


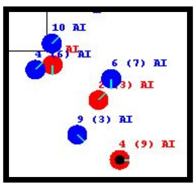
Figure 1: TSG Simulator



①Player giving pass



②Timing above the player's head



③Player receiving pass

Figure 2: Flow of the through pass

2.2 Approach

▪ approach①

If the player has held the ball for 3 frames or more after receiving the ball, it is a pass, otherwise it is not considered a pass.

The pass judgment function is determined by the player ID. This compares the ID of the player who received the ball with the ID of the player who had the ball before. If the ID is different, it is judged as Pass, and if the ID is the same, it is not considered as pass. Observing the situation in which the ball is over the player's head with a simulator, it was found that the ball overlaps the player for 1 frame or 2 frames when passing the player (Figure 3). This means that the ball has finished passing the player at least 3 frames later. Therefore, the number of frames during which the player overlaps the ball is counted, and if it is less than 2 frames, the variable holding the ID of the player who has newly received the ball is not updated and it is not considered as a pass.

Whether or not the player and the ball overlap is determined by a function called "findpwbID()". First, define a variable called "ball_coord" that stores the position of the ball for each frame.The function called "GetplayerByposition()" takes the ball position(x,y) as v, the player position(x,y) as v1, and the range defined as the player as R_2. Then, compare the original player's range R_2 and the dist, which is the range obtained by subtracting the ball position from the player, and if the dist is smaller, get the ID of the player who has the ball, otherwise return NULL. From the above, it is possible to determine whether the player is overlapping with the ball.

```
int findpwbID(gsim::Game const &frame, int teamID)
{
  gsim::gl::Point2d ball_coord = frame.GetBall().GetShape().C;

  const gsim::Player * player =
frame.GetTeams().first[teamID].GetPlayerByPosition(ball_coor
d);

  return player != NULL ? player->GetPlayerID() : -1;
}
```

```
const Player * GetPlayerByPosition(const gl::Point2d &pos)
const
{
  math::Vec2d v = math::make_Vec2d(pos.X, pos.Y);

  const Player *ret = NULL;

  for (std::size_t i = 0, count = m_Players.size(); i < count; ++i)
  {
  math::Vec2d v1 =
math::make_Vec2d(m_Players[i].GetShape().C.X,
m_Players[i].GetShape().C.Y);

  double R_2 = m_Players[i].GetShape().R;
  R_2 *= R_2;

  double dist = math::get_norm_squared(v1 - v);
  if (dist < R_2)
    {
     ret = &m_Players[i];
     break;
    }
  }

  return ret;
}
```

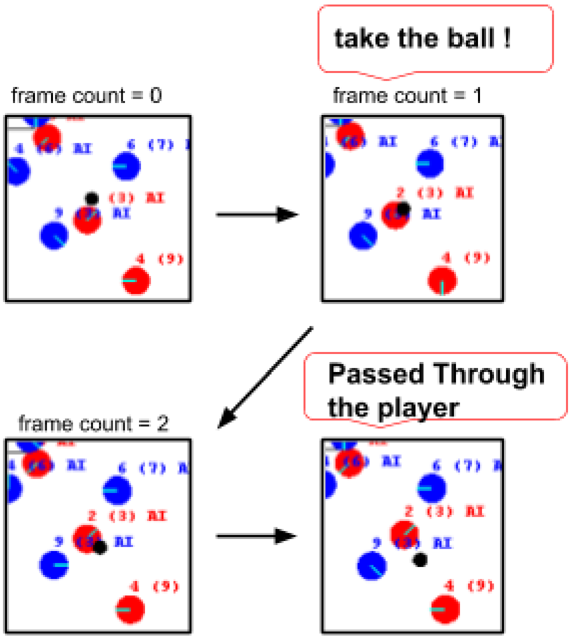


Figure 3: Count of frames when passing though

▪ approach②

The approach I tried to solve the problem is a method of making a pass judgment by changing the direction of the ball.

The direction of the ball is defined as eight ranges, with the range of -180 degrees to 180 degrees divided by 45 degrees. First, the moving distance is calculated from the ball position(x,y) with respect to the field in the current frame and the ball position(x,y) with respect to the field in the immediately preceding frame. And the angle of the ball can be obtained from the arctangent of the moving distance(x,y).

```
gsim::Interface::Action::Direction angleToDirection(double
angle)
{
  // angle is in radians
  angle *= 180 / M_PI;

  if (angle >= -22.5 && angle < 22.5) return
gsim::Interface::Action::directionRight;
  if (angle >= 22.5 && angle < 45 + 22.5) return
gsim::Interface::Action::directionRightDown;
  if (angle >= 45 + 22.5 && angle < 90 + 22.5) return
gsim::Interface::Action::directionDown;
  if (angle >= 90 + 22.5 && angle < 135 + 22.5) return
gsim::Interface::Action::directionDownLeft;
  if (angle >= 135 + 22.5 || angle < -22.5 - 135) return
gsim::Interface::Action::directionLeft;
  if (angle >= -22.5 - 135 && angle < -22.5 - 90) return
gsim::Interface::Action::directionLeftUp;
  if (angle >= -22.5 - 90 && angle < -22.5 - 45) return
gsim::Interface::Action::directionUp;

  return gsim::Interface::Action::directionUpRight;
}
```

In the case of a pass through a player, the ball direction does not change. Therefore, the pass judgment is not performed until the direction of the ball changes after the player receives the ball on the simulator, and the pass is judged when the direction of the ball changes (Figure 5). Conversely, if the direction of the ball does not change, it is determined that it is not a pass, and the variable that holds the ID of the player who has the ball newly is not updated as in Approach①. From the above, I thought that the problem of the pass that goes over the player’s head would be solved.

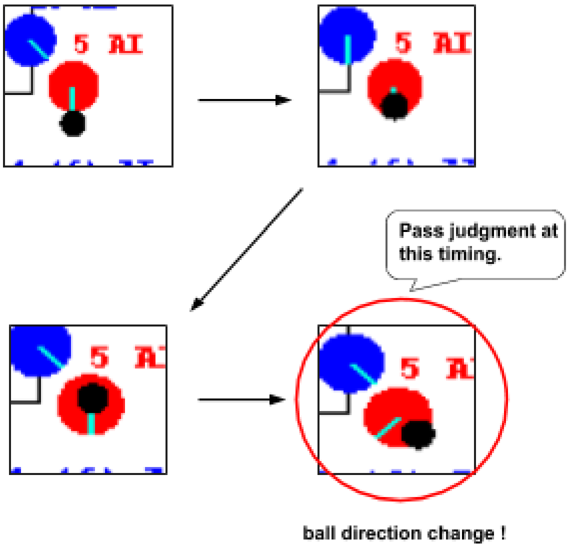


Figure 5 : Timing of pass judgment

2.3 Analysis

Analysis of the timing at which the ball direction changes indicates that the direction of the ball changes depending on the direction of the player after receiving the ball. The direction of the ball also changes if the player passes the ball to another player immediately after receiving the ball. However, there was an exception where the direction of the ball did not change for a while after the player received the ball. The problem occurs in situations where the player is running in the same direction as the ball. In this case, the player who receives the ball keeps running without changing the direction, so the pass judgment will be delayed.

To solve this problem, I created a buffer. This buffer is used to keep storing the frame number, the number of the passed player and the identification ID from when the player receives the ball until the ball changes direction. Even if the ball direction does not change and the pass judgment is delayed, the simulator can perform pass judgment at the correct timing by going back to the buffer information. I thought that using this buffer could solve the problem mentioned above.

3 Result

table 1 shows the precision value of each approach. The precision value indicates the accuracy of the number of passes recognized by the simulator that were passed in an actual soccer match.

table 2 shows the recall value of each approach method. The recall value is the percentage of the simulator's pass recognition that could have been predicted to be a pass in an actual soccer match. The lower the recall value is, the more often the simulator incorrectly recognizes what should be judged as pass.

The correct number of passes for the soccer match used in this study is 582.

・result of approach①

As a result of trying Approach II, 541 passes were recognized. With this method, the simulator incorrectly recognized 41 passes. Of these 41 mistakes, the simulator recognized it as a pass, but in an actual soccer game it was not a pass twice. And the simulator recognizes that it is not a pass, but in an actual soccer game, it has 39 passes.

From these data, the precision value was 0.99 and the recall value was 0.93.

・result of approach②

After trying Approach②, 569 passes were recognized. With this method, the simulator incorrectly recognized 12 passes. Of these 12 mistakes, 12 were recognized as pass by the simulator but were not pass in actual soccer match. From these data, the precision value was 0.97 and the recall value was 1.00.

Tabele 1: Precision

Subjects	Precision
Approach①	0.99
Approach②	0.97

Table 2: Recall

Subjects	Recall
Approach①	0.93
Approach②	1.00

4 Discussion

After trying approach①, 541 passes were recognized. In other words, the simulator incorrectly recognized 41 passes. Of these 41 mistakes, there were two that were determined to be pass by the simulator but were not actually pass. The cause of this mistake is that the ball can rarely take more than 3 frames as it passes through the player. In the simulator, it was judged that it was not a pass, but actually it was a pass 39 times. With this method, the pass performed in 3 frames or less cannot be recognized. Therefore, it missed the pass judgment performed in 3 frames or less and made 38 mistakes.

As a result of trying approach②, 569 passes were recognized. In other words, the simulator incorrectly recognized 12 passes. All of the 12 mistakes were judged as pass by the simulator but were not actually pass. This is because the ball curves and passes through the player (Figure 6). At that time, since the direction of the ball changes, it is judged as pass.

Go through the players
while drawing a gentle curve

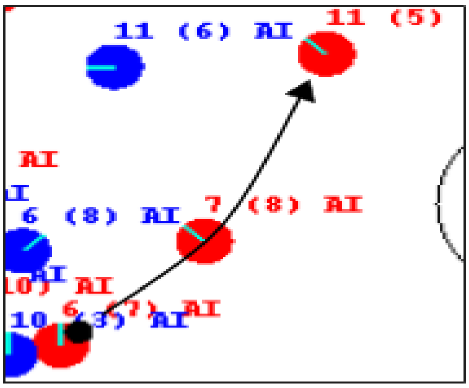


Figure 6 : Example of a ball drawing a curve

From the above, the precision value of approach① is 0.99, the recall value is 0.93, the precision value of approach② is 0.97, and the recall value is 1.00. In addition, approach① always waits for 3 frames before the pass judgment, so the pass judgment timing may be delayed. However, approach② is more flexible than approach① because it can determine the pass at the right time.

Comprehensively, it was found that approach② could simulate an actual soccer game. However, approach① solves many of the problems of passing the ball through the player, resulting in higher precision than approach②. In order to set the precision value of approach② to 1.00, it is necessary to solve the problem that the ball passes through the player while curving.

5 Conclusion

In this study, I simulated a data set of an actual soccer game with a game engine and analyzed whether the judgment of whether or not it passed was closer to the flow of the actual game. I used two approaches to correct the wrong pass judgment. The first is to wait 3 frames after the player receives the ball before making a pass decision. The second method is to make a pass judgment based on whether the direction of the ball has changed since the player received the ball. The precision value of approach① is higher, but the precision value of approach② is as high as 0.97. On the other hand, the recall value of approach① was as low as 0.93, where as the recall value of approach② was 1.00, which was perfect. In addition, the timing of pass judgment is more accurate in approach② than in approach①.

Therefore, approach② can perform a simulation closer to an actual soccer game.

However, the ball may change direction and pass through the player, causing false pass recognition. So this method still needs some improvement.

Acknowledgement

Thanks to Professor Mozgovoy and Mr Khaustov for helpful advice and continuous discussions in writing this paper. And I am grateful to the members of the lab for providing us with the research facility and soccer game engine system, how to use it, and game data.

References

[1] V. Khaustov, M. Mozgovoy.Teaching Automated Software Testing with Appium and Soccer Simulator. North Ossetian State University Bulletin, 2017 , vol. 4, pp. 124-127.

[2] V. Khaustov, G. M. Bogdan, and M. Mozgovoy, "Pass in Human Style: Learning Soccer Game Patterns from Spatiotemporal Data. Proceedings of the 2019 IEEE Conference on Games, London, UK, 2019, pp. 777-778.