Scientific Research Publishing

World Journal of Mechanics, 2019, 9, 163-176
http://www.scirp.org/journal/wjm
ISSN Online: 2160-0503
ISSN Print: 2160-049X

Offside Detection System Using an Infrared Camera Tracking System

Esteban Lopez, Peter E. Jenkins*

Mechanical Engineering Department, University of Colorado, Denver, USA
Email: *peter.jenkins@ucdenver.edu

How to cite this paper: Lopez, E. and Jenkins, P.E. (2019) Offside Detection System Using an Infrared Camera Tracking System. *World Journal of Mechanics*, **9**, 163-176.
https://doi.org/10.4236/wjm.2019.96011

Received: May 29, 2019
Accepted: June 25, 2019
Published: June 28, 2019

Abstract

This paper describes an experimental offside detection system that will be capable of detecting offside passes during a game of soccer. Soccer is the world's most popular and most televised sport. In recent years, FIFA has implemented goal line technology in order to end controversial goals/missed goals during high profile competitive matches. The most contentious aspect of the sport is the offside rule and its many controversial calls or lack of calls. Sometimes the linesmen cannot see the passage of playing fast enough to make a correct decision. Being similar to goal line technology, people have requested offside technology to help the linesmen and to reduce the number of incorrect offside calls in a game. This paper describes a working offside detection system that can accurately detect offside passes. Positional data was exported from a VICON infrared motion tracking camera system and a MATLAB script was written so that it can analyze the positions of the players and the ball and determine if a pass was offside.

Keywords

Soccer, Motion Simulation

1. Introduction

The purpose of this project is to design and develop an experimental offside detection system that will be capable of detecting offside passes during a game of soccer. Soccer is the world's most popular and most televised sport. Every year, thousands of players from all across the world compete in some of the most competitive and prestigious leagues. In recent years, FIFA [1], soccer's governing body, has implemented goal line technology in order to end controversial goals/missed goals during high profile competitive matches. This has been a step in the right direction, however, there are still many issues in the world of soccer.

The most contentious aspect of the sport is the offside rule and its many controversial calls or lack of calls. There have been countless instances when a player is wrongfully ruled offside after scoring a game changing goal. There have also been many cases in which a player has scored a game winning goal and the player was not ruled to be offside when the play should have been stopped. Incorrect offside calls have the potential to ruin games. Sometimes the linesmen, the referees in charge of judging offside calls, cannot see the passage of playing fast enough to make a correct decision. There are many cases in which a ball is played too quickly for anyone to make the correct call. Being similar to goal line technology, people have called out for offside technology to help the linesmen and to reduce the number of incorrect offside calls in a game. That is why the objective of this project is to create a working offside detection system that can accurately detect offside passes. Positional data will be exported from the system and a MATLAB script will be written so that it can analyze the positions of the players and the ball and determine if a pass was offside.

2. Methods

1) The Offside Rule

In order to understand how this offside detection system is going to function, one must understand the offside rule first. The offside rule is defined as follows [2]: "A player is offside when he/she is nearer to the opponent's goal line than the second to last opponent when the ball is played/passed from one player to the player that is nearest to the goal line. Being level with the second to last opponent does not constitute being offside, neither does being level with both the last two opponents if they happen to be in line". When judging the condition of being "nearer", only a player's head, torso, legs, and feet are taken into consideration. The player's arms do not count. A player cannot be offside in his/her own half of the pitch regardless of where he/she is positioned in relation to the ball or members of the opposing team. In order for a player to be considered offside, he/she must be involved in active play. This can be accomplished in two ways. One is to be interfering with the play, or if the player is interfering with an opponent.

2) VICON System Specifications & Setup

A VICON motion tracking camera system [3] and software was used for the development of this system. The College of Arts & Media currently has a VICON system in the Digital Animation Center (DAC). This VICON system consists of 20 Bonita 10 infrared tracking cameras and they are placed along the edges of the room. The cameras all face the center of the room where an 18' × 14' space is designated for digital motion tracking. Each Bonita 10 camera has a maximum frame rate of 250 fps. The lens has an operating range of 13 meters and a maximum field of vision that is 70.29˚ × 70.29˚. The DAC normally uses a VICON software called BLADE for their animation and motion capture needs. Fortunately, VICON also provides a software called Tracker which tracks the

positions of rigid bodies and exports that data as a file, or in real time if needed. VICON was generous and provided a free month-long trial of their Tracker software for this project. The Bonita 10 cameras track rigid bodies via the use of reflective markers. In order to create a rigid body in the system, the reflective markers need to be grouped into unique clusters [4]. Each pattern must be unique, otherwise the cameras will confuse one cluster for another and combine or remove desired rigid bodies. Figure 1 shows some of the reflective markers used in this project.

Once the reflective markers have been put into unique clusters, the next step is to strap them onto the desired rigid bodies. In this project, the objects that need to be tracked are each player's legs, torso, and head. There are four players involved in this simulation, so a total of four heads, four torsos, and eight legs need to be tracked. Figure 2 shows an example of how the markers were strapped onto the players.

Once each player has been strapped with their designated markers, the next step is to use the Tracker software to create the rigid bodies. Once the cameras have been calibrated and set, the software will recognize each marker and each unique pattern and create a rigid body. Each rigid body can be labeled so that one can keep track of which rigid body belongs to each player. Figure 3 shows how each rigid body is represented in the software for each player and the ball.

Additionally, the ball that will be used during this project also needs to be tracked by the VICON camera system. Since the reflective markers that are used to track the players are spheres, they cannot be used to track the ball. The reflective markers would keep the ball from rolling. The solution to this problem was

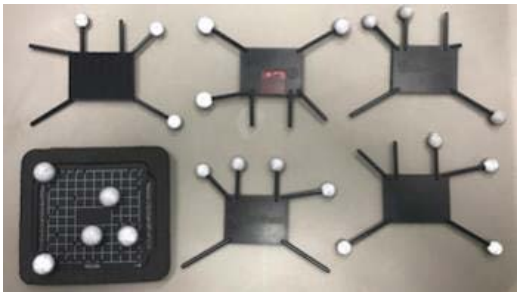

Figure 1. Reflective markers for rigid body tracking.

Figure 2. Reflective marker set up.

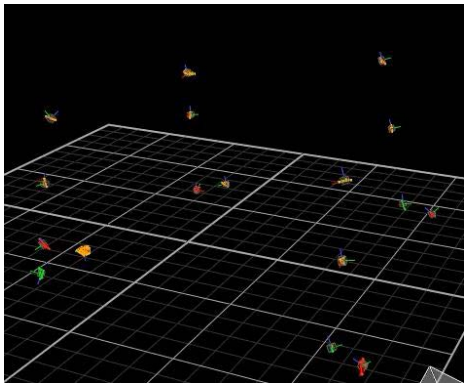

Figure 3. Rigid bodies as shown on VICON Tracker software.

to use OptiTrack reflective tape. The reflective tape was cut into small circles and placed in unique patterns across the ball so that the software could identify the ball as its own rigid body. Figure 4 shows the ball and the reflective tape used for tracking.

3) Single Pass Simulation

In this project, simulations were conducted with four players. Players 1 and 2 were part of Team A. Team A was the attacking team making the passes. Players 3 and 4 were part of Team B. Team B was the defending team setting the offside line. Figure 5 shows the position of both teams before and after the single pass was made.

It can be seen that the only object that moves is the ball. This was a way to control the offside line in order to ensure that they algorithm correctly identified the type of pass. In this situation, the ball is at rest and is passed only once. This simulation was conducted twice. Once for an onside pass and once for an offside pass. In the first pass, player 1 passes the ball to player 2. Both players are behind players 3 and 4, and therefore onside. In the second pass, player 2 is in front of players 3 and 4 and therefore in an offside position. Figure 6 shows the loop in the algorithm that was used to determine the frames in which the ball was moving and shows the first 22 frames.

The loop runs through all the position changes in the X-direction of the ball and finds any frame where the ball has traveled more than 10 millimeters. Once it has found these frames, it places them in the vector shown in Figure 6. The first frame in the vector is where the pass was made. The rest of the algorithm would then find the player's position at this frame and determine if the pass was onside or offside. This report will not go into further detail about the methods used for the single pass scenarios. This report will instead have an in depth look at the methods used for the final running simulation [5].

4) Final Running Simulation

Once the code had been created for a single pass, the next step was to modify the algorithm and run a simulation where all players are constantly changing positions, thus changing the offside line. Figure 7 shows an example of the changes in player position for the final simulation.

Figure 4. Ball and its reflective markers.

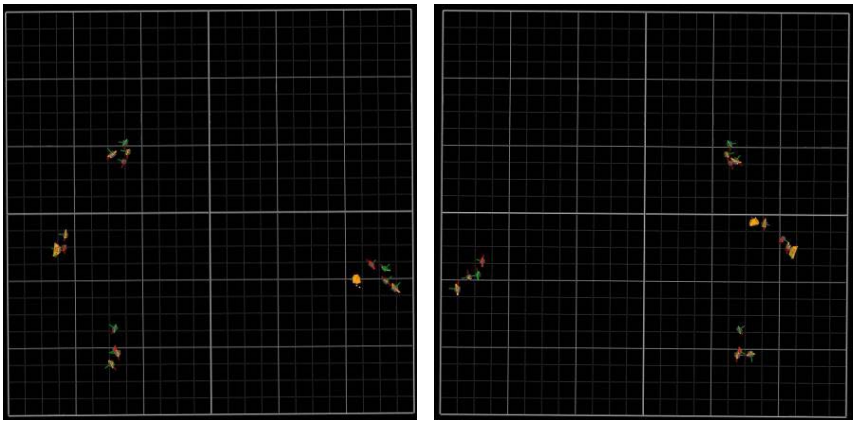

Figure 5. Static trial player positions before & after pass.

```
%% Onside Forward Pass
%%% Analyzing the Ball

%Calculate Row Deltas
D_Ball = diff(Ball);
Deltas_Ball = abs(D_Ball);

k = find(Deltas_Ball(:,4)>10);
z = Deltas_Ball(k,4);

movingFrames = [];
index = 1;
for i=1:(length(k)-1)
    delta = k(i+1)-k(i);
    if delta<2
        movingFrames(index) = k(i);
        index = index + 1;
    end
end

movingFrames = movingFrames';
%A Pass is started at movingFrames(1) at this point in the code
%Now need to compare the positions of all players at the index/frame...
%when pass was made. i.e if pass made at frame 466. Find position...
%of all players at frame 466 and compare.
```

movingFrames

123x1 double

	1	2
1	157	
2	158	
3	159	
4	160	
5	161	
6	162	
7	163	
8	164	
9	165	
10	166	
11	167	
12	168	
13	169	
14	170	
15	171	
16	172	
17	173	
18	174	
19	175	
20	176	
21	177	
22	178	

Figure 6. Loop to determine moving frames.

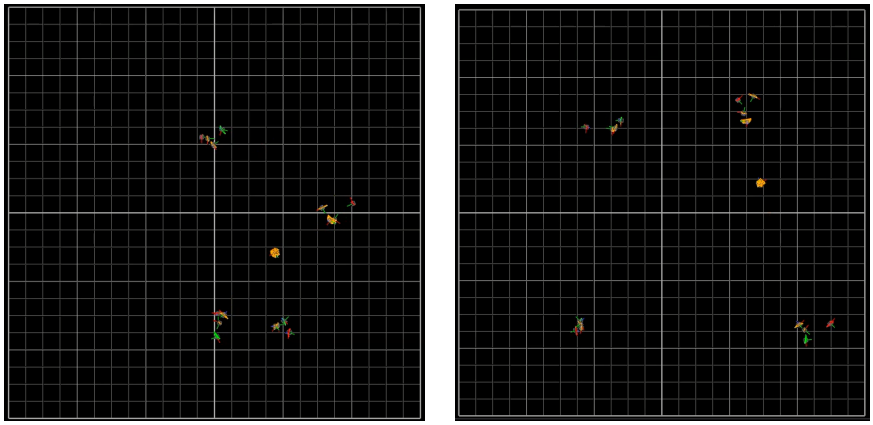

Figure 7. Running simulation player positions changing with time.

This running simulation is the final run that is to be analyzed by the algorithm. For this project, the running simulation lasted 2 minutes and 9 seconds. The code was altered so that it could read all passes sequentially. In order for the algorithm to detect passes, the positional data needed to be loaded onto MATLAB. The data for each rigid body was exported as an excel file. Figure 8 shows the way the data was formatted for all rigid bodies.

The software tracks six different types of positional data for each rigid body. Columns one through three are the rotational positions in the X, Y, and Z directions. These are measured in radians. Columns four through six are the translational positions in the X, Y, and Z directions. These are measured in millimeters. This project will only focus on columns four through six for each rigid body since the translational data is what will be analyzed. In this simulation, the ball was passed under three different conditions. The first type of pass that was observed was when the ball was at rest and was then kicked. In order to determine when the ball was kicked, the code looks at the changes in the X-direction. If the changes in the X-direction went from less than 10 mm per frame to greater than 10 mm per frame, then the ball was considered to be in motion, and thus passed at that specific frame. Figure 9 shows the changes in the X-direction for the ball at frame 2944.

The majority of the passes happened under these conditions. Figure 10 shows how the code was written to capture the frames where these pass conditions were met.

The second type of pass that was observed was when the ball was passed vertically. In this scenario, the code needed to read the changes in the Y-direction for the ball. The same conditions were applied as in the first type of pass. If the changes in the Y-direction went from less than 10 mm per frame to greater than 10mm per frame, then the ball was considered to be in motion, thus passed at that specific frame. Figure 11 shows the changes in the Y-direction for the ball.

Only one pass fell under this criterion and that was at frame 561. Figure 12 shows how the code was written to determine the frame where this pass condition was met.

		Global Angle Player_3_Head: Player_3_Head					
Frame	Sub Frame	RX	RY	RZ	TX	TY	TZ
		rad	rad	rad	mm	mm	mm
1	0	-0.420662	0.539482	4.0564	697.985	-1718.96	1613.23
2	0	-0.397319	0.521966	4.04231	701.481	-1716.14	1613.43
3	0	-0.407936	0.531558	4.04906	705.444	-1714.86	1613.23
4	0	-0.39644	0.514076	4.043	709.124	-1711.97	1613.28
5	0	-0.406556	0.516764	4.04654	712.853	-1710.84	1612.92

Figure 8. Exported positional data for player 3 head first five frames.

	1	2	3	4	5	6
2932	0.0012	0.0018	0.0025	0.0610	0.0320	0.2031
2933	5.5000e-04	0.0024	5.8300e-04	0.1090	0.1640	0.0281
2934	0.0044	1.0000e-05	0.0019	0.0280	0.2130	0.1217
2935	0.0024	9.6000e-04	0.0045	0.1870	0.2850	0.2043
2936	0.0011	4.0000e-05	7.7900e-04	0.0070	0.0010	0.0414
2937	0.0011	0.0018	0.0016	0.0960	0.2620	0.0774
2938	0.0012	0.0030	0.0036	0.0860	0.2530	0.0159
2939	0.0021	0.0055	0.0028	0.0060	0.3590	0.3641
2940	0.0018	0.0020	0.0088	0.1220	0.6420	0.0041
2941	0.0016	0.0023	0.0024	0.1680	0.3130	0.0936
2942	2.5000e-04	0.0055	2.1100e-04	0.0920	0.3540	0.2038
2943	1.0000e-04	0.0099	0.0145	0.2730	0.9080	0.0635
2944	0.0318	0.0365	0.0288	11.6370	6.0090	3.6835
2945	0.0730	0.1185	0.0129	24.9890	8.1310	1.8572
2946	0.0723	0.0985	0.0070	24.2270	9.2310	1.8648
2947	0.0725	0.1002	0.0233	24.8710	9.9680	0.1990
2948	0.0852	0.0507	0.0213	24.5110	10.1580	1.0405
2949	0.0759	0.0673	0.0022	24.7050	11.0640	0.6678
2950	0.0691	0.0654	0.0026	25.3650	11.5280	1.0138
2951	0.0937	0.0511	0.0293	24.6610	9.6710	1.8980
2952	0.0784	0.0429	0.0103	24.7840	11.1140	2.6512
2953	0.0836	0.0431	0.0161	24.6250	10.7820	3.3800

Figure 9. Column 4 shows ball changes in X-direction at frame 2944 for pass Type 1.

```
%% Passes
%%% Analyzing the Ball

%Frame rate
t = 129; %seconds (run time of simulation)
fps = Frames/t;
spf = t/Frames; %seconds per frame

%Calculate Row Deltas
D_Ball = diff(Ball);
Deltas_Ball = abs(D_Ball);
%k = find(Deltas_Ball(:,4)>10);

%%%%3 scenarios where passes are made%%%
%%%Scenario 1 - Ball is at rest and then it is passed%%

Passes1 = [];
index = 1;
for i=1:(length(Deltas_Ball)-1)
    if (Deltas_Ball(i,4)<10 && Deltas_Ball(i+1,4)>10 && Deltas_Ball(i+2,4)>10 && Deltas_Ball(i+3,4)>10 && Deltas_Ball(i+4,4)>10 && Deltas_Ball(i+5,4)>10)
        Passes1(index) = i+1;
        index = index + 1;
    end
end
Passes1 = Passes1';
```

Figure 10. Pass type 1 code for determining relevant frames.

	1	2	3	4	5	6
550	0.0042	0.0091	0.0024	1.0030	0.0200	0.0260
551	0.0174	4.9100e-04	3.4000e-04	0.4040	0.3200	0.0870
552	2.2000e-04	7.7800e-04	0.0080	0.9820	0.4000	0.2220
553	0.0060	0.0032	3.3000e-04	1	0.3000	0.1190
554	0.0125	0.0115	0.0161	1.7350	0.2700	0.1800
555	0.0024	0.0027	0.0056	0.7820	0.9800	0.0740
556	0.0061	0.0082	0.0016	1.2710	0.2200	0.0650
557	0.0103	0.0014	0.0073	0.2160	0.4200	0.0390
558	0.0242	0.0068	2.8000e-04	0.0230	0.0500	0.5350
559	0.0048	0.0023	0.0012	0.9080	0.4500	0.0250
560	0.0062	0.0026	0.0018	0.5910	0.7900	0.2580
561	0.0541	0.0466	0.0026	2.1030	13.9800	0.5970
562	0.0285	0.0544	0.0041	5.4150	28.8100	0.0150
563	0.0423	0.0862	0.0192	5.2470	28.4900	1.4840
564	0.0514	0.0852	0.0128	4.9480	28.4100	0.2110
565	0.0389	0.0859	0.0124	5.0970	28.4600	0.4200
566	0.0284	0.0841	0.0092	5.4350	28.9700	1.3770
567	0.0561	0.1063	0.0094	5.0590	27.9300	1.9270
568	0.0776	0.1027	0.0101	3.4800	28.6500	0.1220
569	0.0279	0.1110	0.0296	5.2140	28.7300	1.3040

Figure 11. Column 5 shows ball changes in Y-direction at frame 561 for pass type 2.

```
%%%Scenario 2 - Ball is passed vertically so need to analyze Delta y
%%%instead of Delta x%%%

Passes2 = [];
index2 = 1;
for j = 1:(length(Deltas_Ball)-1)
    if (Deltas_Ball(j,5)< 1 && Deltas_Ball(j+1,5)>10 && Deltas_Ball(j,2)<.003)
        Passes2(index2) = j+1;
        index2 = index2 + 1;
    end
end

Passes2 = Passes2';
```

Figure 12. Pass type 2 code for determining relevant frame.

The final type of pass that was observed was when the ball had already been passed and was passed again by the receiving player while the ball was still in motion. In this scenario, the ball was never at rest between passes and thus needed to be analyzed in a different way. For this scenario, the velocities of the ball at each frame were determined. If the changes in the X-direction were greater than 10 mm per frame, but there was a sign change in the velocities, then the frames where the sign changes occurred were considered a pass. **Figure 13** shows the changes in the X-direction for the ball at frame 2304.

There is a sign change in the velocities at frame 2304 (**Figure 14**). This means that a pass was made at this frame. **Figure 15** shows how the code was written to calculate the velocities at each frame and how the sign changes were found.

Finally, once pass frames had been isolated, various loops were created in the algorithm that would compare the positions of each player's legs, torso, and head at those frames. A final loop was created to compare those positions between the attacking players and the defending players. **Figure 16** shows how the player's positions at all relevant frames were found.

The final section in the algorithm then determined if the pass was onside or offside. **Figure 17** shows the final loops in the code that determined offside passes.

3. Results

The results for the single pass simulations can be seen on **Table 1**.

The results determined from the algorithm match the results that were obtained by visually confirming the pass from the simulation videos that were recorded from the software.

The running simulation is more complex than the single pass simulations. In order to determine the number of passes and their status, the frames where the passes were made needed to be determined. **Table 2** shows the number of passes and their corresponding frames.

There were 45 passes in total throughout this simulation. However due to glitches in the system as well as gaps in the data due to camera limitations, only 43 out of the 45 passes were recognized by the cameras. This is further explained in the Discussion section of this report. The missing passes did not affect the

	1	2	3	4	5	6
2296	0.0202	0.0679	0.0040	12.0100	2.9770	7.1770
2297	0.0229	0.2066	0.0189	13.8700	4.2330	11.6390
2298	0.0351	0.2045	0.0156	15.9900	3.1270	11.4400
2299	0.0193	0.0993	0.0122	15.9300	4.7620	7.0960
2300	0.0439	0.0904	0.0094	16.1900	3.9970	6.4670
2301	0.6299	0.0700	0.7086	55.4400	17.8960	32.0290
2302	0.0792	0.1349	0.0197	15.0100	1.5640	0.7700
2303	0.0715	0.1571	0.0723	12.4900	3.1340	0.2990
2304	0.0167	0.0334	0.0070	19.1200	1.8270	2.2880
2305	0.0718	0.1268	0.0452	36.1600	6.0230	2.5560
2306	0.0323	0.1060	0.1390	33.1600	2.4410	2.0390
2307	0.0415	0.1446	0.1451	31.5400	2.8470	0.5460
2308	0.0592	0.1138	0.1212	32.4200	0.6880	0.8540
2309	0.0691	0.1274	0.1290	31.6600	1.2480	1.7510
2310	0.1152	0.1336	0.1642	29.8500	2.6560	0.9710
2311	0.0888	0.1467	0.1136	31.2200	0.3010	3.5010
2312	0.0581	0.1643	0.0935	31.0600	0.6410	4.5100
2313	0.1358	0.1736	0.1073	29.2000	0.3300	1.0740

Figure 13. Columns 4 shows ball changes in X-direction at frame 2304 for pass type 3.

	1
2296	-1.2072e+03
2297	-1.3942e+03
2298	-1.6073e+03
2299	-1.6013e+03
2300	-1.6274e+03
2301	-5.5728e+03
2302	-1.5088e+03
2303	-1.2555e+03
2304	1.9219e+03
2305	3.6348e+03
2306	3.3332e+03
2307	3.1704e+03
2308	3.2588e+03
2309	3.1824e+03
2310	3.0005e+03
2311	3.1382e+03
2312	3.1221e+03
2313	2.9352e+03

Figure 14. Shows the calculated velocities for the frames around frame 2304.

```
%%%Scenario 3 - Ball is passed immedietly after receiving ball%%
%%%Ball never stops rolling and is passed from on direction to the next%%%
%%%Need to check velocities for sign changes%%%

v1 = D_Ball(:,4)/spf; %mm/s

Passes3 = [];
index3 = 1;
for k = 1:(length(v1)-3)
    if (v1(k)<0 && v1(k+1)<0 && v1(k+2)>1340 && v1(k+3)>0) %&& v1(k+4)>0 && v1(k+5)>0 && v1(k+6)>0 && v1(k+7)>0)
        Passes3(index3) = k+2;
        index3 = index3 + 1;
    end
end

Passes3 = Passes3';
```

Figure 15. Pass type 3 code for determining relevant frames.

Table 1. Results for the single pass simulations.

Onside Static Pass		Offside Static Pass	
Frame	Status	Frame	Status
466	Onside	157	Offside

```
%% Analyzing Players Positions at Pass

P1_H_Value = [];P2_H_Value = [];P3_H_Value = [];P4_H_Value = [];
P1_LL_Value = [];P2_LL_Value = [];P3_LL_Value = [];P4_LL_Value = [];
P1_RL_Value = [];P2_RL_Value = [];P3_RL_Value = [];P4_RL_Value = [];
P1_T_Value = [];P2_T_Value = [];P3_T_Value = [];P4_T_Value = [];
ind = 1;
for i=1:length(Passes)
    %Player 1
    P1_H_Value(ind) = P1_Head(Passes(i),4);
    P1_LL_Value(ind) = P1_LL(Passes(i),4);
    P1_RL_Value(ind) = P1_RL(Passes(i),4);
    P1_T_Value(ind) = P1_T(Passes(i),4);

    % Player 2
    P2_H_Value(ind) = P2_Head(Passes(i),4);
    P2_LL_Value(ind) = P2_LL(Passes(i),4);
    P2_RL_Value(ind) = P2_RL(Passes(i),4);
    P2_T_Value(ind) = P2_T(Passes(i),4);

    % Player 3
    P3_H_Value(ind) = P3_Head(Passes(i),4);
    P3_LL_Value(ind) = P3_LL(Passes(i),4);
    P3_RL_Value(ind) = P3_RL(Passes(i),4);
    P3_T_Value(ind) = P3_T(Passes(i),4);

    % Player 4
    P4_H_Value(ind) = P4_Head(Passes(i),4);
    P4_LL_Value(ind) = P4_LL(Passes(i),4);
    P4_RL_Value(ind) = P4_RL(Passes(i),4);
    P4_T_Value(ind) = P4_T(Passes(i),4);

    ind = ind+1;
end
P1_H_Value = P1_H_Value';P2_H_Value = P2_H_Value';P3_H_Value = P3_H_Value';P4_H_Value = P4_H_Value';
P1_LL_Value = P1_LL_Value';P2_LL_Value = P2_LL_Value';P3_LL_Value = P3_LL_Value';P4_LL_Value = P4_LL_Value';
P1_RL_Value = P1_RL_Value';P2_RL_Value = P2_RL_Value';P3_RL_Value = P3_RL_Value';P4_RL_Value = P4_RL_Value';
P1_T_Value = P1_T_Value';P2_T_Value = P2_T_Value';P3_T_Value = P3_T_Value';P4_T_Value = P4_T_Value';
```

Figure 16. Code for determining player positions at relevant frames.

```
%% Determining Onside/Offside Pass

A_Attacking = [A1 A2];
A_Offside = [A3 A4];
Offside_Line = [];
ind2 = 1;
for i=1:length(A_Offside)
    Offside_Line(ind2) = max(A_Offside(i,:));
    ind2 = ind2 + 1;
end

Offside_Line = Offside_Line';


V_final = [];
ind3 = 1;
for i=1:length(Passes)
    V_final(ind3) = v1(Passes(i));
    ind3 = ind3 + 1;
end
V_final = V_final';

Pass_Decision = {};
ind4 = 1;
for i=1:length(Offside_Line)
    if (max(A_Attacking(i,:)) > Offside_Line(i) && V_final(i) > 0)
        Pass_Decision{ind4} = {'OFFSIDE'};
        ind4 = ind4 + 1;
    elseif (max(A_Attacking(i,:)) < Offside_Line(i) || V_final(i) < 0)
        Pass_Decision{ind4} = {'Onside'};
        ind4 = ind4 + 1;
    end
end

Pass_Decision = Pass_Decision';
Pass_Numbers = [1:length(Passes)]';
T = table(Pass_Numbers, Pass_Decision);
```

Figure 17. Final loops for determining offside passes.

Table 2. Number of passes & corresponding frames their corresponding frames.

Pass	Status	Pass	Status	Pass	Status	Pass	Status	Pass	Status
1	108	11	2944	21	5687	31	8771	41	12,037
2	329	12	3206	22	5957	32	9591	42	12,320
3	561	13	3446	23	6178	33	10,039	43	12,420
4	903	14	3723	24	6398	34	10,347		
5	1178	15	4016	25	6612	35	10,574		
6	1354	16	4342	26	7105	36	10,854		
7	1631	17	4621	27	7569	37	11,120		
8	1869	18	4907	28	7836	38	11,362		
9	2144	19	5218	29	8237	39	11,612		
10	2304	20	5485	30	8468	40	11,781		

results of this project. Once the number of passes and their corresponding frames had been identified, the algorithm was able to determine whether each pass was onside or offside. Table 3 shows the final results of the running simulation for each pass. Table 4 shows the results from the visual confirmation of each pass and is favorably compared to the simulation results given in Table 4.

4. Conclusion

In conclusion, the offside detection system prototype has successfully identified onside and offside passes. Table 3 and Table 4 in the results section are identical. This means that the algorithm accurately determined the status of all 43 passes in the simulation. The success of the prototype is a major step towards the advancement of technology in soccer.

5. Discussion

Even though the prototype was a success, there were some limitations to the design. The biggest limitation to the design was the fact that the Tracker software was not meant to track a large number of rigid bodies at once. Since each rigid body had to be comprised of a unique pattern of reflective markers, it became difficult to have enough different patterns for each rigid body as the number of rigid bodies increased. Additionally, the software would sometimes confuse one rigid body for another if two players got to close to one another. If one player ran past another, there were instances where the markers on one player's leg were too close to another player's leg, thus creating one rigid body that the software could not recognize. As a result, there are gaps in the data that was exported from the software. Fortunately, these gaps did not occur at the instances where the passes were made. However, moving forward, this could be an issue if the gaps were to occur during important sections of the simulations.

Gaps in the data did not only occur when multiple rigid bodies got too close to one another. They also occurred whenever the ball was in motion. Since the

Table 3. Pass results from algorithm.

1 Pass_Numbers	2 Pass_Decision
1	Onside
2	Onside
3	Onside
4	Onside
5	Offside
6	Onside
7	Onside
8	Onside
9	Onside
10	Offside
11	Onside
12	Onside
13	Onside
14	Offside
15	Onside
16	Onside
17	Onside
18	Offside
19	Onside
20	Offside
21	Onside
22	Onside
23	Offside
24	Onside
25	Offside
26	Onside
27	Onside
28	Onside
29	Offside
30	Onside
31	Offside
32	Onside
33	Offside
34	Onside
35	Offside
36	Onside

Continued

37	Offside
38	Onside
39	Offside
40	Onside
41	Onside
42	Onside
43	Onside

Table 4. Pass results from visual confirmation of simulation recordings.

Pass	Status	Pass	Status	Pass	Status	Pass	Status	Pass	Status
1	Onside	11	Onside	21	Onside	31	OFFSIDE	41	Onside
2	Onside	12	Onside	22	Onside	32	Onside	42	Onside
3	Onside	13	Onside	23	OFFSIDE	33	OFFSIDE	43	Onside
4	Onside	14	OFFSIDE	24	Onside	34	Onside		
5	OFFSIDE	15	Onside	25	OFFSIDE	35	OFFSIDE		
6	Onside	16	Onside	26	Onside	36	Onside		
7	Onside	17	Onside	27	Onside	37	OFFSIDE		
8	Onside	18	OFFSIDE	28	Onside	38	Onside		
9	Onside	19	Onside	29	OFFSIDE	39	OFFSIDE		
10	OFFSIDE	20	OFFSIDE	30	Onside	40	Onside		

software needs to see all reflective tape marks on the ball in order to establish it as a rigid body, the bottom portion of the ball did not have any reflective tape on it. This resulted in gaps in the data whenever the ball rolled since the bottom portion of the ball would end up on top while in motion. When this happened, the cameras would momentarily lose track of the ball for a few frames. The cameras would detect the ball once it had completed a rotation and all pieces of tape were visible yet again. These gaps in the data affected two of the passes during the simulation. Frames 9175 and 12,202 saw two short and quick passes between players 1 and 2. Since there were gaps in the data, the software did not track these passes. Fortunately, both of these passes were onside and happened at a short distance where they did not affect the results of the simulation. All offside passes were accurately tracked and were not affected by the gaps in the data. Additionally, there were some movement restrictions throughout this project. The VICON system was located inside a computer lab/classroom, therefore all passes needed to remain on the ground. The ball needed to be passed at a reasonable speed so that no equipment or students would get injured.

Moving forward there are many improvements that could be made to this experimental offside detection system. The main improvement that can be made is to improve the camera system. The VICON system works great and the infrared

tracking is accurate. However, in a real game setting this system will be impossible to use. The infrared tracking will simply not work in an outdoor environment. In order for the cameras to detect the reflective markers, there needs to be zero sunlight in the room and all other reflective surfaces need to be covered up. Additionally, in a real game setting it is not practical to have spherical trackers on the player's bodies. One solution is to utilize reflective tape on the players so that they can move as they normally would. However, this would only be practical in an indoor setting. The best way to move forward is to adopt a camera system that uses artificial intelligence and machine learning to detect color and track objects based on color. There have been numerous research studies done within the last 10 years and the technology is there. In a real game setting these cameras can be used outdoors and would be able to track the colors of each team's jerseys. The VICON system does however work accurately and effectively in terms of the scope of this project. This project is the foundation for what is meant to be a radical change in the world of soccer. Offside technology is years away from being perfected, however this project has demonstrated that it can be done accurately and effectively given the proper resources. The purpose of this project has been completed and the main objective has been met. The VICON infrared camera system can accurately detect offside passes. This technology can be used to aid the referees and officials to fairly influence the outcome of soccer games.

Acknowledgements

1) The authors would like to thank the University of Colorado Denver Digital Animation Center.

2) The support of VICON was appreciated.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] The Hidden Technologies at the 2018 FIFA World Cup, June 4 2018. https://football-technology.fifa.com/en/blog/hidden-technologies-at-the-2018-fwc

[2] Laws of the Game-FIFA, 2015/2016. https://img.fifa.com/image/upload/datdz0pms85gbnqy4j3k.pdf

[3] VICON Motion Capture System. https://www.vicon.com/motion-capture

[4] Cortes, N., Blount, E., Ringleb, S. and Onate, J.A. (2011) Soccer-Specific Video Simulation for Improving Movement Assessment. *Sports Biomechanics*, **10**, 22-34. https://doi.org/10.1080/14763141.2010.547591

[5] Bodenheimer, B., Rose, C., Rosenthal, S. and Pella, J. (1997) The Process of Motion Capture: Dealing with the Data. In: *Computer Animation and Simulation*, Springer-Verlag, Berlin, 3-18. https://doi.org/10.1007/978-3-7091-6874-5_1