Research Article

Real Time Robot Soccer Game Event Detection Using Finite State Machines with Multiple Fuzzy Logic Probability Evaluators

Elmer P. Dadios¹ and Soo ho Park²

¹ Department of Manufacturing Engineering and Management, De La Salle University, 1004 Manila, Philippines

² Department of Computer Science, Korea Advanced Institute of Science and Technology, Taejeon 305-701, South Korea

Correspondence should be addressed to Elmer P. Dadios, elmer.dadios@dlsu.edu.ph

Received 10 February 2009; Revised 6 August 2009; Accepted 13 October 2009

Recommended by Kok Wai Wong

This paper presents a new algorithm for real time event detection using Finite State Machines with multiple Fuzzy Logic Probability Evaluators (FLPEs). A machine referee for a robot soccer game is developed and is used as the platform to test the proposed algorithm. A novel technique to detect collisions and other events in microrobot soccer game under inaccurate and insufficient information is presented. The robots' collision is used to determine goalkeeper charging and goal score events which are crucial for the machine referee's decisions. The Main State Machine (MSM) handles the schedule of event activation. The FLPE calculates the probabilities of the true occurrence of the events. Final decisions about the occurrences of events are evaluated and compared through threshold crisp probability values. The outputs of FLPEs can be combined to calculate the probability of an event composed of subevents. Using multiple fuzzy logic system, the FLPE utilizes minimal number of rules and can be tuned individually. Experimental results show the accuracy and robustness of the proposed algorithm.

Copyright © 2009 E. P. Dadios and S. H. Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

For the past years, robot soccer games are played without human intervention; however, the games are refereed by a real person [1, 2]. If it is meant to be a real robot soccer game, then the human referee is better replaced by a machine. This machine referee is somewhat similar to the robot intention extraction problem [3].

Among the decisions made by a human referee for a robot soccer game, giving score or goal point is the most critical because it directly decides the winner of a game. For that matter, to detect goalkeeper charging fouls is very crucial in order to decide and give correct score points to a team [4].

A charging foul decision is a special case of detecting collisions between robots because the information given to the system is inaccurate and insufficient [5, 6]. The decision is dependent on the positions and the actions of the robot defensive goalie and the robot attacker/shooter. Based on the Federation of International Robot-Soccer Association (FIRA) rules [2], a goalkeeper charging foul is called when an attacker or a shooter directly pushes the goalkeeper away

from its current position. The attacker robot is permitted to push the goalkeeper only through a ball in between them. However, even if there is a ball between the attacker and the goalkeeper, pushing the goalkeeper passed the goal line is considered a charging foul.

The multiple robot tracking and object collision detection has been widely researched [7–12]. Literature review reveals that the exact shapes and coordinates of objects must be given so that the knowledge of the physics can be utilized [8, 13–15]. However, in this study to use the former method of collision detection is not possible because based on FIRA rules, the information given to the system is only the team color and the individual robot identity color using colored patches placed on top of the robot. The shape and size of the colored patches are open for a team to decide. Thus the real time information available to the robot soccer event referee system is inaccurate and insufficient because of the limitations of the camera vision system [5, 16, 17]. This makes this problem very challenging and interesting to solve.

In this paper, finite state machines with multiple fuzzy logic probability evaluators algorithm are used in developing

and implementing an event referee for a robot soccer game. A state machine is assigned to detect a specific event. It uses the probabilities of event occurrence calculated by fuzzy logic probability evaluators (FLPEs) and threshold crisp probability values to discern if the events certainly happened. The process is administered by a main state machine that initiates other state machines based on situational conditions of the game such as the actual position and velocities of the robots and the ball. The system developed has extendible structure for using other state machines modules so that new features of the machine referee can be easily added and adopted.

This paper is organized as follows. Section 1 discusses the scope and direction of the research. Section 2 discusses the mechanics of robot soccer identification and collision detection. Section 3 discusses the design architecture of the proposed event referee system for the robot soccer game. Section 4 discusses the fuzzy logic probability evaluator algorithm to calculate the crisp value of the type of events that triggers. Section 5 presents the experiment results with analysis and discussions. Finally, Section 6 presents the conclusion and recommendations.

2. The Robot Soccer Identification and Collision Event Detection

The mechanics of the Microrobot Soccer Tournament are formulated by FIRA [1, 2]. The actual size of the robot is 7.5 cubic centimeter [18]. Each robot is identified by a camera vision system through the two colored patches on the top part of its body [16]. Each team has an assigned patch color while each robot's identity has a designated patch color. The team color is either yellow or blue. The individual robot identity color can be of any color except yellow or blue. The size and shape of each team and individual robot colored patches is not given provided that it should not exceed 3.5 square centimeters [5].

2.1. Collision Event of Two Robots. The robot collision event can be divided into 3 subevents. The first event is when a robot moves close toward another robot. The second event is when a robot strikes another robot resulting in its decreased speed, and the third is when both robots move in two general ways as shown in Figure 1. The case in Figure 1(a) is the two robots bounced. It is called "Hitting." The other case in Figure 1(b) is the two robots move together like one object as they try to push each other. It is called "Pushing."

It is important to note that the goalkeeper will move toward the ball when the ball is near the goal to prevent the ball from going inside the goal. Other robots will also show the same behavior to attempt to possess the ball, especially when opponent robots are close by. In addition, the goalkeeper usually moves parallel to the goal line to effectively protect the goal area. When the goalkeeper moves perpendicular to the goal line with considerable speed, then it gives a certain degree of evidence of a collision. This is illustrated in Figure 2.

FIGURE 1: Two possible cases of collisions: "Hitting" and "Pushing."

FIGURE 2: (a) The usual parallel to the goal line movement of a goalkeeper and the perpendicular movement at collision. (b) The expected movement of a robot when the ball is near and the movement at collision.

2.2. Types of Events. A frame is the captured image of the physical scene at a moment, and the frame information includes the coordinates and speeds of the robots. The type of events in a robot soccer game can be divided into two groups based on the number of frames needed to detect the event. These groups are as follows:

- (1) the events that can be detected with one frame information; for example, two objects are in the goal area at the same time; this can be detected with the coordinate information of two objects in one frame;
- (2) the events that can be detected using more than one frame information; for example, a stalemate for 10 seconds; this can be detected only after checking all the frames for 10 seconds.

Also, the events can be categorized according to the number of subevents:

- (1) primitive events: the events that do not have subevents; for example, the ball is near the robot.
- (2) composite events: the events that have subevents; for example, the collision event has 3 subevents as described in Section 2.1.

2.3. Motions and Orientations of Two Robots during Two Consecutive Frames. Figure 3 shows the motions of two robots before and after capturing two consecutive frames. In here, *Robot A* moved downward and *Robot B* moved upward. There are four important parameters that are needed to describe the dynamics of the two robots.

FIGURE 3: Robot orientation before and after two consecutive frames.

- (1) SV represents the distance traveled by *Robot A*. The origin is the center of the team's colored patch of *Robot A original position*, and the destination is the center of the team's colored patch of *Robot A new position*.
- (2) DV1 represents the distance between *Robot A new position* and *Robot B new position*.
- (3) DV2 represents the distance between *Robot A original position* and *Robot B new position*.
- (4) θ_d represents the angle between SV and DV2. Note that the smaller value of θ_d means that *Robot A* moves toward *Robot B*. The DV and SV of other objects like the ball can be defined in the same manner.

3. The Robot Soccer Game Event Referee System

The architecture of the robot soccer game event referee system developed in this research is shown in Figure 4. This is composed of two main parts: (1) the vision system, and (2) the decision making system. The vision system is the same as the one used in the remote-brainless vision-based soccer robot system [7]. It is composed of a digital camera overlooking the robots and connected to a host computer. The vision system provides the frame information like the coordinates and direction of the robots [16–18].

The decision making system will detect events like pushing, hitting, stalemate, ball shot, and others by inspecting the frame information. The fuzzy logic probability evaluators (FLPEs) give the probability of the occurrence of events. The state machines (SMs) use a threshold crisp probability value. For example, if the threshold value of an event is 80%, the event will be regarded as it happened when the probability of its occurrence calculated by the FLPE is above 80%. The detail about an FLPE is discussed in Section 4.

3.1. Multiple State Machines. The Main State Machine (MSM) shown in Figure 4 is the heart of the robot soccer game referee system. The MSM uses a vision system to align its action with the State Machines (SMs). It initiates the State Machines (SMs) based on heuristic rule using $E0$. For example, if an opponent robot is within 20 cm distance from

FIGURE 4: The Robot soccer game event referee system architecture.

the goalkeeper robot, then the MSM will initiate the $S0$ to detect possible goalkeeper charging. The $S0$ is initiated only when the event it detects has the probability to occur in order not to waste the resources of the host computer. The MSM also makes final decisions, for example, if an SM notifies a “goalkeeper charging” while another SM reports a “goal shot”, the MSM ignores the “goal shot” event and takes the “goalkeeper charging” event.

It is easy to modify or extend the event referee system by adding a state machine for a new situation and updating the rule base of the main state machine to accommodate this condition. Figure 5 shows a sample Finite State Machines for detecting goalkeeper charging and stalemate. The events like $E1, E2, \dots, En$ are detected by Fuzzy Logic Probability Evaluators (FLPEs) and threshold crisp probability values. For example, if the threshold value is set at 80%, an event with the probability higher than 80% will be regarded as if it actually happened. The threshold values may be set differently for each event or the same for all events. The details and explanation of the actions taken from Figure 5 are described in Table 1 with the following definitions.

$FLPE(n)$ is the fuzzy logic probability evaluator for a given specific number n .

$P(En)$ is the probability of the event En calculated by $FLPE$ for En at frame n .

TPn : is the threshold Probability for En .

4. Fuzzy Logic Probability Evaluators (FLPEs)

The $FLPEs$ are designed to find probabilities of the occurrence of events. There are basic $FLPEs$ for primitive events and composite $FLPEs$ for complex events. A composite $FLPE$ consists of $FLPEs$ for each sub-event of the complex event. The probability of an event to occur is defined in fuzzy linguistic terms shown in Figure 6. These terms are used for all other $FLPEs$. Since we are less concerned with low probability, it is defined as VL (Very Low) with larger area. The descriptions of each $FLPE$ are given in Table 2. The input variables to some $FLPEs$ are explained as well.

FIGURE 5: Sample architecture of finite state machines for detecting goalkeeper charging and stalemate.

TABLE 1: The states and actions taken for Figure 5.

State	Action
MSM	(1) Get information from vision system and initialize a state. (2) Give final decision
S0	(1) Use $FLPE4$ to calculate $P(E1)$ (2) If $(P(E1) > TP1)$ then go to S1. Else go to MSM and get next frame information.
S1	(1) Use $FLPE3$ to calculate $P(E2)$ and $FLPE8$ for $P(E2')$ using $P(E1)$ from S0 as the input of $FLPE3$ and $FLPE8$. (2) If $(P(E2) > TP2)$ then go to S2 Else If $(P(E2') > TP3)$ then go to S3 Else go to MSM and get next frame information.
S2	(1) Use $FLPE11$ to calculate $P(E3)$ (2) If $(P(E3) > TP3)$ then go to S4. Else go to S3
S3	(1) Use $FLPE10$ to calculate $P(E5)$ (2) If $(P(E5) > TP5)$ then notify the MSM for occurrence of "goalkeeper charging" foul, Else go to MSM and get next frame information.
S4	(1) If the goalkeeper passes the goal line, then notify the MSM for occurrence of "goalkeeper charging" foul, Else go to MSM and get next frame information.
S5	(1) Use $FLPEs$ to calculate $P(E7)$, $P(E8)$, and $P(E9)$ (2) If $((P(E7) > TP7) \text{ or } (P(E8) > TP8)) \text{ and } (P(E9) > TP9)$, then go to step 3, else exit (3) If $(P(E6) > 10 \text{ sec})$, then notify the MSM for occurrence of "stalemate," Else go to MSM and get next frame information.

TABLE 2: Detail description of Fuzzy Logic Probability Evaluators (FLPEs).

Number	Name (description)	Type
FLPE1	<i>Probability Combiner</i> It gives the probability of the event occurrence composed of two subevents by combining probabilities of two subevents (See Section 4.2 and Figure 7)	
FLPE2	<i>Collision Detector</i> It gives the probability of the collision between two robots. It is composed of FLPE3 and FLPE8 (See Section 4.2 and Figure 7)	Composite
FLPE3	<i>“Hitting” type Collision Detector</i> It gives the probability of the “Hitting” type collision between two robots. It is composed of FLPE4, FLPE5, FLPE6, and FLPE7	Composite
FLPE4	<i>A robot moves close toward another robot</i> It gives the probability of the primitive event occurrence that a robot moves close toward another robot using inaccurate and insufficient data (See Section 4.3 and Figure 7)	Basic
FLPE5	<i>A robot swiftly decreases its speed near another robot</i> Same as FLPE4 with the event mentioned in the name. This FLPE uses 2 input variables namely, (1) variation of the speed of a robot decreasing its speed (S_v) and (2) Distance between two robots (D_r)	Basic
FLPE6	<i>A goalkeeper moves perpendicular to the goal line</i> Same as FLPE4 with the event mentioned in the name. This FLPE uses 2 input variables Namely, (1) angle of a goalkeeper movement (θ_g) and (2) Speed of a goalkeeper movement (S_g).	Basic
FLPE7	<i>A robot moves not to the ball when the ball is near</i> Same as FLPE4 with the event mentioned in the name. This FLPE uses 3 input variables Namely, (1) probability of the ball being very near to the robot (P), which will be given by FLPE10, (2) angle difference between the SV of a robot and the DV of a robot to the ball (θ_{db}), and (3) speed of a robot movement (S_r).	Basic
FLPE8	<i>“Pushing” type Collision Detector</i> It gives the probability of the “Pushing” type collision between two robots. It is composed of FLPE4, FLPE5, and FLPE9	Composite
FLPE9	<i>Two robots move together</i> Same as FLPE4 with the event mentioned in the name. This FLPE uses 2 inputs, namely, (1) angle difference between the SV of two robots (θ_{dr}) and (2) Speed difference of the two robots (S_d).	Basic
FLPE10	<i>The ball is near to a robot</i> There are two modes for FLPE10. The first is to check if the ball is very near to a robot as to inspect if a robot hits the ball. The second is to check if the ball is considerably near to a robot as to inspect if the ball is coming to a goalkeeper.	Basic
FLPE11	<i>The ball is between two robots</i> It gives the probability that the ball is between two robots. It is used to check the location of the ball when “pushing” happens. If the ball is between two robots, “pushing” is legal if it is outside the goal area	Composite
FLPE12	<i>The ball position is on or beyond the goal line</i>	Basic
FLPE13	<i>The ball is almost not moving</i>	Basic
FLPE14	<i>The ball position is outside the goal area and inside the soccer field</i>	Basic

4.1. *Probability Combiner (FLPE1)*. The probability combiner FLPE1 is used to combine probabilities of subevents to get the probability of a complex event. FLPE1 basically gives the probability of an event composed of two subevents. By using FLPE1 consecutively, the probability of an event composed of many subevents will be calculated. The FLPE1 is used to find the probability of composite events composed of two subevents. It uses 2 crisp probabilities for each sub-event of a composite event as inputs and gives the crisp

output probability for the composite event using the center of gravity defuzzification.

The probability of a sub-event that actually happened could have low value due to the inaccurate data from the vision system. Also, the behavior of objects is continuous. Therefore, the probability of a sub-event could be low if the frame is taken after a moment that could be characterized as the sub-event. The subevents of an event occur in sequence or at the same time. Though the probability of one sub-event

FIGURE 6: Membership functions for the fuzzy logic probability evaluators (VH: very high, H: high, M: medium, L: low, VL: very low).

TABLE 3: Fuzzy associative memory matrix (FAM) for FLPE1.

		P1 (Probability of sub-event 1)				
		VH	H	M	L	VL
P2 (Probability of sub-event 2)	VH	VH	VH	H	M	M
	H	VH	H	H	M	L
	M	H	H	M	L	VL
	L	M	M	L	L	VL
	VL	M	L	VL	VL	VL

is very low, if the probability of the other sub-event is very high, then the probability of the event composed of these two subevents can be said as medium. It is to think that the high probability of a sub-event gives certain evidence that the other sub-event is triggered despite its low probability. The FAM on Table 3 gives all these rules based on the algorithm in Table 4. This can be interpreted as follows:

IF P1 is VL (very low) and P2 is VH (very high) **THEN** the probability of the event composed of subevents 1 and 2 is M (medium).

4.2. Collision Probability Evaluator (FLPE2). The FLPE2 calculates the probability of a robot's collision with another robot given the frame information of two robots such as speed vector and distance vector (refer to Section 2.3). This is composed of two FLPEs that are also composed of other FLPEs. The first is to detect the "Hitting" type collision (FLPE3) and the second is to detect the "Pushing" type collision (FLPE8).

In Figure 7, FLPE3 has two types of "Hitting" conditions. The left part detects the event when an opponent robot hits the goalkeeper and the goalkeeper moves perpendicular to the goal. The right part detects the event when a robot hits another robot and prevents that robot from going to the ball. This applies not only for the goalkeeper but also to all other robots. Whichever has the higher probability between the two parts will be selected for the "Hitting" type collision event.

FLPE8 shown in Figure 7 uses three subevents FLPE4, FLPE5, and FLPE9 to detect the "Pushing" type collision event. Note that FLPE1 is again used to calculate the final fuzzification of these three subevents. Finally, the higher probability value between two probabilities of "Hitting" and "Pushing" will be selected as the probability of a collision event between two robots.

From Section 2.1, there are three subevents of collision. The first sub-event is a robot moving close toward another robot in one frame, and the other two subevents happen

together at the next frame. Thus, to get the probability of a collision at frame N , the output of FLPE4 for the first sub-event should be taken from the previous frame of frame N while the outputs of the other two subevents are taken from the current frame N .

4.3. Probability Evaluator for an Event That a Robot Moved Closer Toward Another Robot (FLPE4). The FLPE4 uses 3 input variables.

- (1) D is the distance between *Robot A new position* and *Robot B new position* (in cm).
- (2) θ_d is the angle between *Robot A new position* and *Robot B new position* based from *Robot A old position* (in degrees).
- (3) S_a is the speed of approaching *robot A* (in cm/frame).

The FLPE4 initially uses inputs D and θ_d to calculate the temporal probability of the event using the membership functions in Figures 8 and 9, and the rules in Table 5. When D and θ_d have small values, it is more probable that a robot moves close toward another robot. Note that due to vision system difficulties, even when the robot is not moving, the value of S_a can be nonzero: it is usually lower than the value of S_a obtained when the robot actually moves. To distinguish this error from the actual movement, the membership function of S_a is defined as shown in Figure 10. The final probability is calculated as the product t -norm of the temporal probability and the membership value of S_a . For example, if S_a is 0, then the membership value is 0 and the final probability becomes 0. If the S_a membership value is 1, then the final probability equals the temporal probability.

5. Experimental Results

In order to prove the robustness of the system, various experiments were conducted. These were done using different types of patches to show that the system is not dependent on a single patch design. Also, the experiments were conducted in various locations and orientations within the playing field. The captured images of continuous frames and the values of FLPEs for various cases are shown in this section. Note that in these experiments, a heuristic value 84% was used for the threshold value for all FLPEs. This crisp value is between the fuzzy linguistic probability terms H (for high) and M (for medium). The abbreviations below are used and are defined to simplify the analysis.

- PO3: is the crisp probability output of FLPE3, for deciding a "Hitting" type of collision.
- PO4: is the crisp probability output of FLPE4, for deciding that a robot moved closer toward another robot.
- PO5: is the crisp probability output of FLPE5, for deciding that a robot swiftly decreased its speed near another robot.
- PO6: is the crisp probability output of FLPE6, for deciding that a goalkeeper moved perpendicular to the goal line

TABLE 4: The heuristic knowledge for building the FAM in Table 3.

- (1) Let the linguistic probability terms VH and H belong to the set *HIGH*, let L and VL belong to the set *LOW*, and let M belong to the set *MID*
- (2) For some elements h_1, h_2 of *HIGH*, m of *MID*, l_1, l_2 of *LOW*, the binary operation ($\bullet$) in *FLPE1* between two elements is defined as
- $h_1 \bullet h_2$ = choose the higher probability term. Example, $VH \bullet H = VH$
  - $l_1 \bullet l_2$ = choose the lower one. Example, $VL \bullet L = VL$
  - $h_1 \bullet m$ = choose the middle or the higher one. Example, $H \bullet M = H$, $VH \bullet M = H$
  - $l_1 \bullet m$ = choose the lower one. Example, $L \bullet M = L$
  - $h_1 \bullet l_1$ = choose the middle one (when there are two middle ones, choose the one near the lower one). Example, $VH \bullet L = M$

FIGURE 7: Multiple fuzzy logic probability evaluator for collision type detection.

FIGURE 8: Membership functions for θ_d (S: small, M: medium, B: big). FIGURE 10: Membership function for S_a (CH: considerably high). FIGURE 9: Membership functions for D (N: near, M: medium, F: far).TABLE 5: Fuzzy associative memory matrix for a part of *FLPE4*.

		D		
		N	M	F
θ_d	S	VH	H	M
	M	H	M	L
	B	M	L	VL

PO7: is the crisp probability output of *FLPE7*, for deciding that a goalkeeper moved away from the ball.

PO8: is the crisp probability output of *FLPE8*, for deciding a pushing type of collision.

PO11: is the crisp probability output of *FLPE11*, for deciding that a ball is between two robots.

PO12: The crisp probability output of *FLPE12*, for deciding that a ball is on or beyond the goal line.

5.1. Goal Decision. Figure 11 shows the four consecutive frames of a real time event of a goal score experiment. In this case, the attacker successfully made a goal without striking the goalkeeper as the goalkeeper was late to defend the goal. From Table 6, the values of hitting collision PO3 and pushing collision PO8 are below the threshold value 84% which means that no collision occurred between the two robots. The attacker robot swiftly decreased its speed as it went nearer the goalie, as shown in the higher value of PO5 at Frame 3. The system correctly declared a “goal score” at the 3rd and 4th frames as shown from the values of PO12. The robustness of the system can be seen from other indicators through smaller values of PO4, PO6, and PO7. Figure 12 shows the summary of this result.

FIGURE 11: Four real time frames to decide goal shot behavior. (a) The attacker moves to the goal area to shoot the ball. (b) The goalkeeper tries to defend. (c) The goalkeeper is late and the attacker reaches the goal area. (d) A "goal shot" is made without any foul.

FIGURE 12: The graphical representation to summarize the robots behavior shown in Figure 11 with a threshold probability value of 84 %.

TABLE 6: The % *FLPE* crisp values of the four frames for Figure 11.

FRAME	PO3(%)	PO4(%)	PO5(%)	PO6(%)	PO7(%)	PO8(%)	PO11(%)	P012(%)
1	27.5	68.2	30.7	61.3	38.1	27.4	85.3	50.5
2	68.9	86.1	75.7	61.7	26.7	27.9	92.2	83.7
3	68.9	65.0	90.2	61.5	26.7	41.5	92.2	94.1
4	37.9	59.1	69.1	52.3	75.8	26.8	64.5	100

TABLE 7: The % *FLPE* crisp values of the six frames for Figure 13.

FRAME	PO3(%)	PO4(%)	PO5(%)	PO6(%)	PO7(%)	PO8(%)	PO11(%)	P012(%)
1	27.7	83.1	26.7	78.3	61.7	27.0	84.6	20.5
2	54.8	88.5	26.7	78.3	61.7	33.3	89.9	58.3
3	69.3	81.7	56.7	91.7	96.7	51.9	90.0	64.1
4	79.9	90.9	72.9	96.7	96.7	80.7	90.0	70.4
5	84.4	91.0	64.4	96.4	96.7	90.0	90.0	79.7
6	83.5	91.1	61.7	96.4	96.7	91.6	90.0	84.1

FIGURE 13: Six real time frames to decide charging foul behavior. (a-b) The attacker is moving close toward the goalkeeper with the ball. (c) Two robots are about to collide with the ball in between them. (d-e) The attacker is pushing the goalkeeper backward with the ball in between them. (f) The goalkeeper is pushed into the goal area (Note that the arrow head in between robots is the ball position).

TABLE 8: The % *FLPE* crisp values of six frames for Figure 15.

FRAME	PO3(%)	PO4(%)	PO5(%)	PO6(%)	PO7(%)	PO8(%)	PO11(%)	PO12(%)
1	27.3	81.6	29.1	78.3	61.7	27.3	81.2	64.5
2	44.9	88.2	51.6	28.3	66.8	29.9	85.5	78.3
3	68.2	84.2	73.0	78.3	61.7	49.9	83.2	81.1
4	85.3	88.9	83.8	61.3	96.7	38.5	90.2	85.4
5	85.6	78.6	90.5	61.6	96.7	88.8	90.9	95.7
6	80.6	30.0	61.7	61.2	96.7	27.2	67.2	100

TABLE 9: The results of repeated experiments using patches in Figure 12 (N: the number of experiments; S: success; F: Fail; H: the number of incidences of “Hitting” type collision; P: the number of incidences of “Pushing” type collision; AP: the average probability values given by the system for each type of collision).

	N	S	F	Success rate (%)	H	AP (%)	P	AP (%)
Front	50	48	2	96	32	90.0	16	90.7
Side	50	47	3	94	38	88.8	9	87.6

5.2. Goalkeeper Charging Decision (Pushing). Figure 13 shows the six consecutive frames of a real time event for a goalkeeper charging foul decision. Table 7 shows the *FLPEs* crisp values for the six frames. At frame 1, the attacker brings the ball towards the goal area with the goalie guarding in front of the goal area. At Frame 2, the attacker moves fast

TABLE 10: The experiment results of goal score decision.

	N	S	F	Success rate (%)	AP (%)
Front	50	50	0	100	55.6
Side	50	50	0	100	43.9

and reaches the goalie position with the ball situated between them. This can be verified from the higher value of PO4 and PO11. At Frame 3, the goalkeeper holds his defense position which can be verified from higher values of PO6 and PO7. At Frame 4, the attacker continuously pushes the goalkeeper towards the goal area but still no foul is called because the ball is in-between them. This can be verified from high values of PO4 and PO11, at the same time lower values of PO3, PO8, and PO12. At Frame 5, the Pushing and

FIGURE 14: The graphical representation to summarize the robots behavior shown in Figure 13 with a threshold probability value of 84%.

Hitting collisions are detected which can be verified from values of PO3 and PO8 that are both above 84%. However, the ball is still in between the robots and the goalkeeper did not pass the goal line which can be verified from the values of PO11 and PO12; hence, goalkeeper charging decision is not yet called. At Frame 6 when the goalkeeper passes the goal line together with the ball, a goalkeeper charging has been decided due to pushing collision that can be verified from the higher value of PO8 and lower value of PO3. Note that PO5 values are still low because the attacker is slow and there is no big difference of the speed when it hits the goalkeeper, but the large values of PO4 and PO7 justify this decision. All of these conditions can be verified at Table 7 and summarized at Figure 14.

5.3. Goalkeeper Charging Decision (Hitting). Figure 15 shows the six consecutive frames of the real time event for goalkeeper charging decision caused by hitting collision. To show the robustness of the system developed, different types of team patches are used for the goalkeeper in this experiment. In here, the attacker comes from the right side of the goal area. From Table 8, at 4th frame, the value of PO3 which is hitting collision is above 84%. Hence, the system decides for a goalkeeper charging even though the ball is between the two robots because the goalkeeper has reached the goal line position. This can be verified from the higher values of PO11 and PO12. Table 9 shows the result of various experiments conducted using this team patch. It can be seen from here that the system is very accurate with a success rate of 96%.

5.4. Stalemate Decision. Figure 16 shows the condition for stalemate decision. In here, the ball is not moving and the

robots are stuck. The FLPEs give low probabilities of robots moving to the ball. After 10 seconds is counted by the computer clock, the referee system calls for a stalemate or a free-ball decision.

5.5. Goal Score Decision. There is no case that the “goal score” is incorrectly declared even though in some cases, the attacker robot passes very near the goalkeeper robot, which can be misjudged as a “goalkeeper charging”. Table 10 shows the experiments results. The AP is the average probability of collision calculated by the fuzzy logic evaluators.

5.6. Discussion and Analysis. The system developed was tested for various situations of collision events to get the machine referee decision. Experiment results showed very high accuracy and robustness of the system. This was due to many indicators used together and applying fuzzy logic approximate reasoning capability for dealing with inaccurate data.

Most failed cases were caused by the vision system. When the robot covered the ball and moved together with the ball, the vision system failed to trace the ball due to the small number of pixels captured. For this, a heuristic technique was used. If an FLPE needed the position of the ball and the vision system cannot locate it, it used the FLPE value of the previous frame. However it was found out that if the vision system loses the trace too early, the referee system failed to detect some events such as the ball was between two robots and the ball was near the goalkeeper. This can be solved using a vision system with high resolution as it may trace the ball better.

FIGURE 15: Six real time frames to decide charging foul behavior (a-b) The attacker is approaching the goalkeeper from the side area (c) Two robots are about to collide (d) The attacker strikes the goalkeeper (e-f) The goalkeeper bounces backward and the ball goes into the goal area (Note that the arrow head in between robots is the ball position).

FIGURE 16: A stalemate decision.

6. Conclusions

An event referee system for microrobot soccer game was developed. Robot goal shots, charging fouls, held ball, and stalemates are samples of events that the system can determine accurately. The algorithms in detecting robot collisions under the condition of inaccurate and insufficient information were presented. The paper introduced fuzzy

logic probability evaluators for multiple state machines as a novel way to calculate probabilities for events decision making. Experiment results gave very high success rate and robustness even though the input information is not accurate and insufficient. This was due to the integration of many indicators together and using fuzzy logic's capability for dealing with uncertainties. Other functions of the machine referee decision can be added and integrated into

the system easily due to its extendible structure design using multiple state machines. With the combination of multimedia functionality, the system will be able to facilitate the Microrobot Soccer tournament game actively in the future. The algorithm developed has extendible structure that can easily be applied to other applications or problems. This can be used for other event detection researches as an alternative to using crisp logic.

The authors conducted other techniques to solve the robot soccer game event detection problem. Experiments like curve fitting and direct heuristic or rule based approach were implemented. However, the results of these techniques were not encouraging because the inputs of the system which are taken from the camera are subject to extreme noise that can vary abruptly. The proposed technique using the weighted sum and cascaded or multiple fuzzy logic probability evaluators is very effective in filtering the noise. For future works the authors will implement this proposed technique in developing humanoid behavior.

Acknowledgments

The authors would like to acknowledge the support provided by the administrators of the De La Salle University and the DLSU Science Foundation that insured the success of this research.

References

- [1] J. H. Kim and D. H. Kim, *Soccer Robotics*, Springer, New York, NY, USA, 2004.
- [2] J. H. Kim, et al., "Taking spirit of science, technology to next generation," in *Proceedings of the 12th FIRA RoboWorld Congress*, San Francisco, Calif, USA, June 2007.
- [3] K. Han and M. Veloso, "Automated object behavior recognition applied to robotic soccer," in *Proceedings of the IJCAI-99 Workshop on Team Behavior and Plan Recognition*, Stockholm, Sweden, 1999.
- [4] S. Park, E. P. Dadios, and J. H. Kim, "Machine referee for the micro-robot soccer game," in *Proceedings of the FIRA RoboWorld Congress*, Singapore, December 2005.
- [5] E. P. Dadios and O. A. Maravillas, "Fuzzy logic controller for micro-robot soccer game," in *Proceedings of the IEEE Industrial Electronics Conference (IECON '01)*, vol. 1, pp. 2154–2159, Denver, Colo, USA, December 2001.
- [6] E. P. Dadios and O. A. Maravillas, "Cooperative mobile robots with obstacle and collision avoidance using fuzzy logic," in *Proceedings of the 17th IEEE International Symposium on Intelligent Control (ISIC '02)*, pp. 75–80, Vancouver, Canada, October 2002.
- [7] E. Maravillas, N. H. Reyes, E. P. Dadios, et al., "Hybrid fuzzy logic strategy for soccer robot game," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 8, no. 1, pp. 65–71, 2004.
- [8] M. Lin and S. Gottschalk, "Collision detection between geometric models: a survey," in *Proceedings of the 8th IMA Conference on The Mathematics of Surfaces*, p. 464, Birmingham, UK, September 1998.
- [9] F. Guedea, R. Soto, F. Karray, and I. Song, "Building intelligent robotics systems with distributed components," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 10, no. 2, pp. 173–180, 2006.
- [10] H. Han, "Adaptive fuzzy control for a class of discrete-time nonlinear systems with state observer," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 10, no. 2, pp. 225–233, 2006.
- [11] Z. Liu, M. H. Ang, and W. K. G. Seah, "A searching and tracking framework for multi-robot observation of multiple moving targets," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 8, no. 1, pp. 15–21, 2004.
- [12] T. Semeon, S. Leroy, and J. P. Laumond, "Path coordination for multiple mobile robots: a resolution-complete algorithm," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 42–49, 2002.
- [13] H. Park and Z. Bien, *A study on intention extraction of soccer object system*, M.S. thesis, Department of EECS, KAIST, Taejon, Korea, 2003.
- [14] A. Witkin, D. Baraff, and M. Kass, "Rigid body dynamics, in a course: an introduction to physically based modeling," in *Proceedings of the 22nd Annual Conference on Computer Graphics (SIGGRAPH '95)*, Los Angeles, Calif, USA, August 1995.
- [15] M. Jamshidi, et al., *Applications of Fuzzy Logic: Towards High Machine Intelligence Quotient Systems*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997.
- [16] N. Reyes and E. P. Dadios, "A fuzzy approach in color object detection," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '02)*, vol. 1, pp. 232–237, Bangkok, Thailand, December 2002.
- [17] S. Barrido and E. P. Dadios, "Online robot tracking using genetic algorithms," in *Proceedings of the 17th IEEE International Symposium on Intelligent Control (ISIC '02)*, pp. 479–484, Vancouver, Canada, October 2002.
- [18] N. Reyes and E. P. Dadios, "Dynamic color object recognition using fuzzy logic," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 8, no. 1, pp. 29–37, 2004.