Fuzzy Logic Collision Avoidance for Autonomous RC Car Follower Utilizing Monocular Camera as Distance Approximator

I. Saleh[1*], W. Rahiman[2]
[1, 2] School of Electric & Electronic Engineering, Universiti Sains Malaysia
[*]corresponding authors: nims14_eee041@student.usm.my

Abstract – *This paper presents the collision avoidance method of a car follower by using fuzzy logic. The distance between the Lead Vehicle (LV) and Follower Vehicle (FV) is approximated using machine vision. Firstly, unit test is performed to check the reliability of the vision system approximation. Once calibrated, the system is validated by integrating the finalized algorithm to the developed prototype. The experiment comprised of testing the capability of the prototype to avoid collision with the lead vehicle when the lead vehicle stops abruptly in two conditions; straight path and curved path. The results show that the prototype was able to avoid collision in most cases and the usage of set classifier improves mean percentage error of distance detection and prevented false trigger of the braking system.*

Keywords: *fuzzy logic, vision system, autonomous rc car*

Article History
Received 31 July 2018
Received in revised form 17 August 2018
Accepted 17 August 2018

I. Introduction

Road injuries and fatalities have been a growing concern in Malaysia. Statistics shows that more than 6,000 motoring fatalities are recorded each year [1]. The number of death due to road accidents has increased from 6,287 in 2006 to 7,152 in 2016 [2]. From January to September 2017, 5,083 deaths were recorded in 400,788 road accidents nationwide [3]. It is also reported that 80.6% of the total accidents were due to human error [4].

There has been significant progress made by countries which adopt a multi-prong approach in tackling road accidents [5]. There are five pillars to the multi-prong approach of road safety namely road safety management, safer roads, safer vehicles, safer road users, and post-crash response.

A. Motivation of Study

While it is not feasible to address these five elements simultaneously, systems that are expected to be achieved on a short-term basis can only be safer vehicle; as those systems can exploit already existing traffic infrastructures.

As proposed by [6], any on-board system for Intelligent Transport System (ITS) applications has to meet these important requirements:

(1) The proposed system must be robust enough to adapt to different conditions and changes of environment, road, traffic, illumination, and weather.
(2) On-board system for ITS applications requires high level of reliability. Therefore, it is important for the system to undergo extensive phase of testing and validation.
(3) For commercial reasons the design of an ITS system has to be driven by strict cost criteria that should not cost more than 10% of the vehicle price. Low power consumptions is desired and the system should not disrupt the car performance.
(4) In order to retain the car design, the system's hardware and sensors have to be kept compact in size.
(5) The system has to be user friendly.

Hence, foreseeing a massive and widespread use of autonomous sensing agents, the use of passive sensors, such as cameras, obtains key advantages over the use of active ones (laser-based sensors). Although machine vision could not exceed human sensing capabilities such as in the absence of fixed illumination, it can, however, help the driver in case of failure, for example in the lack of concentration or drowsy conditions.

The highlight of this paper is to test the developed Remote Controlled (RC) Car follower's ability to avoid collision by using monocular camera as distance approximator; which act as an input to fuzzy logic decision making.

B. Geometry Model

Fig. 1 shows the geometry model in determining distance explaining the representation of image plane in real world (physical dimension) and its corresponding pixel coordinate perceived by vision system [7]-[8].

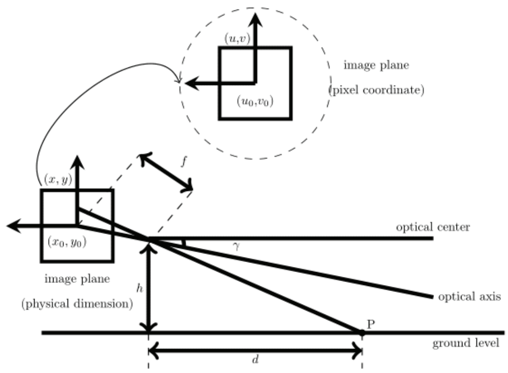

Fig. 1. Geometry model in determining distance

From the figure, P as the target object, d is the distance between the monocular vision sensor and the target object, γ is the camera angle, and h is the elevation height of the monocular vision sensor from ground level. Based on the geometry relationship the distance d could be calculated as by Equation (1).

$$d = h/\tan(\gamma + \arctan((y - y_0)/f)) \tag{1}$$

Suppose that f is the focal length of the monocular vision sensor, (u_0, v_0) is the camera coordinate intersection point of optical axis and image plane, and the physical dimension of a pixel corresponding to the x-axis and y-axis on the image plane are δx and δy respectively, the projection of P on the image plane (image coordinate) (x, y) can be represented as:

$$\begin{cases} x = (u - u_0)dx \\ y = (v - v_0)dy \end{cases} \tag{2}$$

The optical point u and v can be represented as Equation (3)-(4):

$$u = \frac{x}{\delta x} + u_0 \tag{3}$$

$$v = \frac{y}{\delta y} + v_0 \tag{4}$$

By assuming (x_0, y_0) as origin:

$$x_0 = y_0 = 0 \tag{5}$$

The final equation for the distance (d) is:

$$d = h/\tan(\gamma + \arctan((v - v_0)/a_y)) \tag{6}$$

where:

$$a_y = f/\delta y \tag{7}$$

The value of u_0, v_0 and a_y can be obtained by solving the camera matrix as shown in Appendix (A). The result is tabulated in Appendix (B.1).

The value of v is the end point of y-coordinate of the object where h is the height in pixel of the object and y_{init} is the starting coordinate for y-axis ($y_{init} = 0$ if there is no image cropping).

$$v = y + y_{init} + h \tag{8}$$

II. Methodology

A. Vision System

In this section the Lead Vehicle (LV) detection using vision system is explained. The raw input of the system is the real-time video acquired using the monocular camera. The resolution of the webcam is set to as Eq. (9), where w_f is the frame width and h_f is the frame height.

$$0 < w_f \leq 320, 0 < h_f \leq 240 \tag{9}$$

The position of Lead Vehicle is tracked using Camshift. Fig. 2 shows how Camshift works [9]. One drawback of a vision system utilizing a monocular camera is its inability to capture the optimal lighting of its surrounding. In the previous study [10], a suitable illumination compensation method was chosen. As a continuation of the study the compensation method will be integrated to the Camshift tracker algorithm. The algorithm will start with a random solution to a problem, loop (iterate) itself so that it can acquire an improved solution by incrementally changing a single element of the solution. If the alteration generates a better solution, an incremental alteration is made to the new solution, repeating until no more adjustments can be found [11]. The smoothed centroid position output (C_x) of the Lead Vehicle extracted from the Camshift tracker will be one of the crisp inputs for the fuzzy system. Next, the fuzzy system will be further discussed.

B. Fuzzy Logic System

Fig. 3 shows the fuzzy system. Firstly, the crisp input(s) will enter the fuzzifier. The fuzzifier consists of designated input membership functions. Once fuzzified, the fuzzy input(s) will enter fuzzy inference block which consists of rules that these input(s) should abide to.

The resultant fuzzy output(s) will then enter the defuzzifier block. Since the fuzzy system follows the Sugeno method, the defuzzifier employs weighted average calculation approach. From the defuzzifier, crisp output is selected from the designated output membership functions.

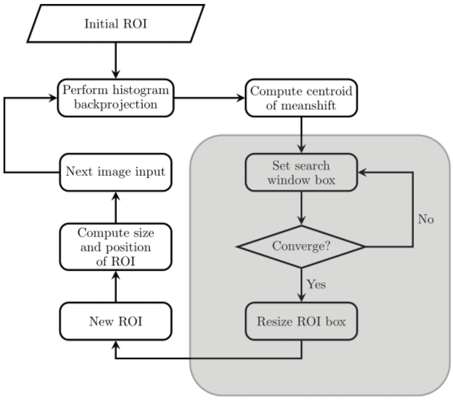

Fig. 2. Camshift Tracker Flowchart

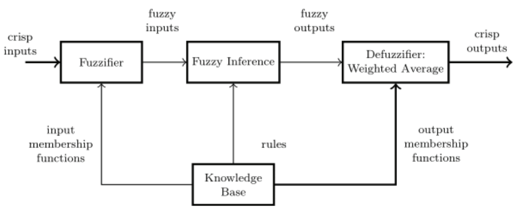

Fig. 3. Fuzzy System

For the implementation of fuzzy logics on embedded controller, zero order type Sugeno fuzzy control is used. The consequent output is in a form of a singleton. Singleton is a fuzzy set with a membership function that is unity at a single particular point on the universe discourse and zero everywhere else. For this system, two crisp inputs are used:

(1) the position of C_x (deviation)
(2) the approximate distance between lead vehicle and follower vehicle (distance)

The consequent outputs would be:
(1) the speed of the motor (speed)
(2) the steering direction of the servo motor (steering)

The fuzzy system was coded in Python based on the functional programming proposed in [12] and [13].

B.1 Fuzzy Sets with Membership Functions

There are two types of fuzzy sets used which are triangle shaped fuzzy sets (Λ) and trapezoidal fuzzy membership functions ($\prod$).

Triangle shaped fuzzy sets are defined as shown in Equation (10); where x is the crisp input, Λ_A is the lower limit of the triangle, Λ_B is the middle limit of the triangle and Λ_C is the upper limit of the triangle.

$$\Lambda\left(x;\underset{ABC}{\Lambda}\right)=\begin{cases}0 & if\ x<\Lambda_A\\ \dfrac{x-\Lambda_A}{\Lambda_B-\Lambda_A} & if\ \Lambda_A\le x\le\Lambda_B\\ \dfrac{\Lambda_C-x}{\Lambda_C-\Lambda_B} & if\ \Lambda_B<x\le\Lambda_C\\ 0 & if\ x>\Lambda_C\end{cases} \tag{10}$$

Trapezoidal shaped membership functions are defined as shown in Equation (11); where $x$the crisp input is and $\prod_{ABCD}$ indicates the vertices of the trapezoid.

$$\prod\left(x;\prod_{ABCD}\right)=\begin{cases}0 & if\ x<\prod_A\\ \dfrac{x-\prod_A}{\prod_B-\prod_A} & if\ \prod_A\le x\le\prod_B\\ 1 & if\ \prod_B<x\le\prod_C\\ \dfrac{\prod_D-x}{\prod_D-\prod_C} & if\ \prod_c<x\le\prod_D\\ 0 & if\ x>\prod_D\end{cases} \tag{11}$$

Table I shows the linguistic variable for the first input; deviation. There are five regions that represent its linguistic values and its consequent pixel ranges which are extreme left (ll), left (l), center (c), right (r) and extreme right (rr).

TABLE I
SELECTED DEVIATION NOTATIONS AND RANGE

Linguistic Variable : Deviation (pixel)			
Notation	Linguistic Value	Range [0,320]	Membership Function
ll	Extreme Left	[0,108]	$\prod$
l	Left	[96,140]	Λ
c	Center	[120,200]	Λ
r	Right	[180,224]	Λ
rr	Extreme Right	[212,320]	$\prod$

Table II on the other hand shows the linguistic variable for the second input; distance. There are three regions that represent its linguistic values and its consequent ranges which are near (d_n), ideal (d_ i) and far (d_ f).

TABLE II
SELECTED DISTANCE NOTATIONS AND RANGE

Linguistic Variable : Distance (cm)			
Notation	Linguistic Value	Range [0,140]	Membership Function
d_n	Near	[0,100]	$\prod$
d_i	Ideal	[90,130]	Λ
d_f	Far	[120,140]	$\prod$

Table III shows the linguistic variables that represent the first output; the speed of the RC car which are stopping the motor/brake (s_b), normal speed (s_n) and accelerate (s_a).

TABLE III
SELECTED SPEED NOTATIONS AND RANGE

Linguistic Variable : Speed (m/s)		
Notation	Linguistic Value	Value
s_b	Brake	0
s_n	Normal	0.8
s_a	Accelerate	1

Table IV shows the linguistic variable that represents the second output; the steering angle of the RC car. There are five regions mapped accordingly which are steer extreme left (str_ll), steer left (str_l), steer center (str_c), steer right (str_r) and steer extreme right (str_rr).

TABLE IV
SELECTED STEERING NOTATIONS AND RANGE

Linguistic Variable : Steering (°)		
Notation	Linguistic Value	Value
str_ll	Steer Extreme Left	50
str_l	Steer Left	70
str_c	Steer Center	90
str_r	Steer Right	110
str_rr	Steer Extreme Right	130

B.2 Fuzzy Sets with Set Qualifiers

In addition to the fuzzy sets introduced in Section B.1, fuzzy set qualifiers called hedges are used.

Hedges are linguistic terms that modify the shape of fuzzy sets [14]. Hedges that are used in this method are slightly and somewhat. The shape of the fuzzy set $m_A(x)$ is represented as shown in Equation (12).

$$hedges = \begin{cases} slightly = m_A(x)^{(1/3)} \\ somewhat = m_A(x)^{(1/2)} \end{cases} \qquad (12)$$

The graphical representation of the fuzzy hedges is shown in Fig. 4. These hedges allow the fuzzy sets to be less rigid than the predefined ranges.

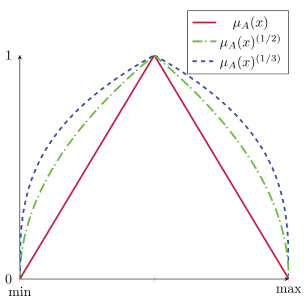

Fig. 4. Fuzzy Hedges

C. Fuzzy Rules

Initially, there are total of 15 fuzzy rules that the system has to evaluate. With the introduction of hedges/fuzzy set classifier, 3 additional rules (rule 16-18) were included. This results in a total of 18 fuzzy rules. All 18 fuzzy rules are shown in Appendix C.

Hypothetically, these additional rules were meant to make the decision making less rigid and prevent frequent signal transitions sent to the actuating component.

D. Development of Prototype

The prototype uses the body of the 1/10 scale Remote Controlled (RC) car. The RC car comes with a built in brushless DC motor and an Electric Speed Controller (ESC). The motor type is Brushless Inrunner 3300KV (3650 size) while the ESC type is 60A Brushless (60A Constant, 85A Burst). With this ESC the motor can be controlled to move in both directions.

D.1 System Architecture

The system architecture is shown in Figure 5. There are two components which are the vision system component and actuating system component. The vision system component is governed by the microprocessor, Raspberry Pi. The image processing sequence is coded in Python with the help of image library OpenCV. There is a touchscreen panel attached to display the guided user interface (GUI). The vision sensor, a webcam is connected to the microprocessor through USB connection.

The actuating system component is governed by a microcontroller, Arduino Mega. The microcontroller is connected to servo motor that will control the steering of the RC car and the ESC, to control the brushless DC motor (BLDC). On the motor a hall sensor is attached and the data is sent to the microcontroller.

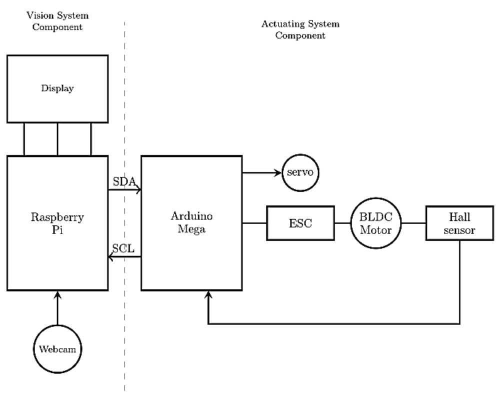

Fig. 5. System Architecture

The fully developed prototype is shown in Appendix D. There are three slots created on the body of the RC car. The bottom tier is reserved for the Electronic Speed Controller (ESC) and Lithium Polymer (Li-Po) battery to power the ESC and consequently the motor. The power supply for both servo motor and microprocessor are slotted to the first tier whereas the microprocessor is slotted to the second tier. The image sensor is mounted on the topmost tier alongside with LCD display for calibration purposes. The prototype will be used to test the algorithm's capability to control the RC car and avoid collision with the Lead Vehicle.

D.2 Power Sources

Table V shows the types of batteries used in developing this prototype. In order to power the ESC and brushless DC motor Lithium Polymer (Li-Po) battery is used. Raspberry Pi needs individual power supply as it is crucial to maintain continuous supply of 5V 2A for the device to work properly. Another power bank with the capacity of 5V 1A is used to power the microcontroller which is Arduino Mega. The servo is powered by Nickel Metal Hydride (NiMH).

TABLE V
TYPES OF BATTERIES USED

Battery	Capacity	Usage
Li-Po	2 Cells, 7.4V	ESC & Brushless DC Motor
Power Bank	5V, 2A	Raspberry Pi & Display
Power Bank	5V, 1A	Arduino Mega
Nickel Metal Hydride (NiMH)	7.2V, 2A	Servo Motor

III. Results & Discussion

A. Experimental Results

In this section the experimental results are presented. Firstly, the unit test is performed by testing the reliability of the distance approximation obtained from the Camshift tracker. Once validated, the improved algorithm is integrated with the prototype and tested for collision avoidance capability.

A.1 Distance Approximator Test

In order to estimate the distance of Lead Vehicle (LV) from Follower Vehicle (FV), Equation (6)) is tested in real time. Fig. 6 shows the plot of distance estimation. By using the equation as it is, it could be observed that there are significant amount of errors. On the plot d_{real} is the actual distance at the particular instances of the experiment.

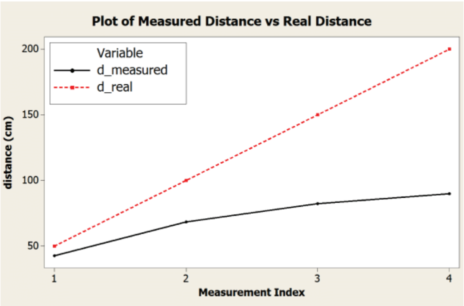

Fig. 6. Distance Plot

New data is tabulated ranging from 50 cm to 200 cm in 10 cm increment. The linear (com_linear), quadratic (com_quad) and cubic (com_cubic) fitted plot are tabulated alongside with real distance and unfitted distance (calc_dist) as shown in Fig. 7. The limit for estimating distance is at 160 cm where the calculated distance reached a plateau. This limit is attributed to the resolution limit of the vision system itself. Nevertheless, it is a useful information when tuning the parameters of the fuzzy logics.

Through further observation from the plot the cubic fitted line plot shows better prediction at near distance but linear fitted line predicts better at far distance. However, cubic fitted plot tends to fluctuate when input distance exceeds the value of the tabulated data. Therefore the chosen fitted line method is linear. The equation for the linear fitted line is shown in Equation 13.

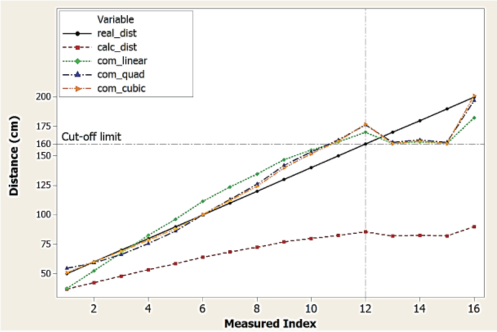

Fig. 7. Connected estimated distance plot

$$d_{\delta} = linear = 63.38 - 1.725\, d_{calc} \qquad (13)$$

A.2 Integrated Test

The finalized distance approximator equation is then integrated to the prototype. The experiment was designed to test the capability of the prototype to avoid collision with the lead vehicle when the lead vehicle stopped under two different conditions namely straight and curved paths. From the result of all 15 trials (5 trials for straight path and 10 trials for curved path) the prototype FV failed only once during curve path. However, it was discovered that the failure was due to tyre misalignment after repetitive testing and not because of false detection.

Without fuzzy set classifier, it was found that there was a delayed response of an average 5 ms per (video) frame. The Follower Vehicle failed to maintain the ideal distance with the Lead Vehicle throughout the path due to this reason. However, when the Lead Vehicle stopped at the end of the path, the Follower Vehicle managed to stop without colliding with the Lead Vehicle. It is observed that with fuzzy set classifier the speed transition was much smoother and FV was able to accurately maintain the ideal distance throughout the path (refer to Table II).

The difference between the apparent distance perceived by the vision system and the real distance between the Lead Vehicle and Follower Vehicle at the end of the path where the Lead Vehicle stopped for each trial is then tabulated as shown in Fig. 8 (straight path) and Fig. 9 (curve path).

For ease of tabulation, ~A represents the algorithm without classifier (15 rules) and ~B represents the algorithm with classifer (18 rules). For both figures, *RealA* and *RealB* represent the measured stopping distance between FV and LV. Meanwhile, *ApparentA* and *ApparentB* represent the perceived stopping distance between FV and LV by the machine vision (calculated by distance approximator).

Table VI shows the percentage tabulation error of the real distance and the apparent distance obtained from the vision system. From the table, high error percentage is recorded when 15 fuzzy rules algorithm (without classifier) is used. However, these values are still within the ideal stopping distance between the Lead Vehicle and Follower Vehicle.

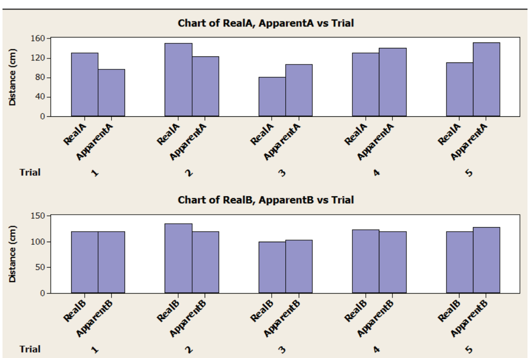

Fig. 8. Comparison of Measured and Perceived Stopping Distance (straight path)

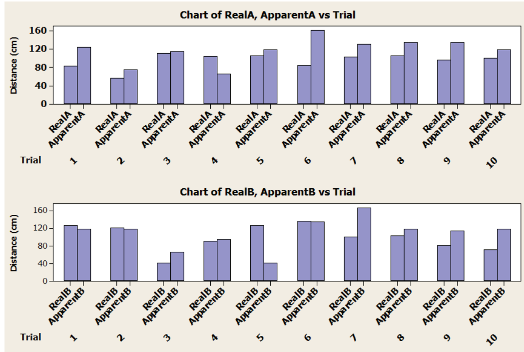

Fig. 9. Comparison of Measured and Perceived Stopping Distance (Curve Path)

TABLE VI
VISION SYSTEM DISTANCE DETECTION ERROR

Path	Mean Error (%)	
	15 rules	18 rules
Straight	22.10	4.21
Curved	34.24	33.98

IV. Conclusion

As a conclusion, the FV prototype managed to avoid collision. There are several limitations that could be highlighted from the findings of the experiments.

(1) The system field of view is limited.
- Due to computational constraint, the vision system resolution is set to 320 x 240 pixels.
- Due to the fixed position of the camera, the follower vehicle could easily lose track of the lead vehicle if the lead vehicle is steering at a sharp corners.

(2) The system fuzzy logic is hardcoded
- The user does not have the privilege to change the fuzzy rules in real-time as the rules were set beforehand based on the experimental data.

(3) Single sensor is not meant for complex navigation
- Due to budget constraints multiple sensors prototype could not be realized

For future works, in order to increase the accuracy of the vision system, a stereo-vision camera could be implemented. The increased in resolution helps expand the field of view of the follower vehicle. Other alternative is to mount the camera on a servo motor so that the camera is able to rotate and lock the target location (LV).

In order for the car to be more intuitive a much more complex algorithm could be introduced (eq: neural network and genetic algorithm).

To top that, other sensors such as laser scanner, line sensor and lux sensor can help tackle the limitations of the vision sensor. The application of laser scanner can help compensate any fluctuating distance estimation and also advantageous when used on uneven terrain. The line sensor will guide the FV to always follow inside the lane and finally, the information from the lux sensor can be utilized so that the vision system can auto-tune itself based on the surrounding ambience. To achieve these, a much more powerful microprocessor is needed. An alternative to Raspberry Pi would be Intel Atom development board.

Appendix

A. Calibration Matrix

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} a_x & 0 & u_0 \\ 0 & a_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

B. Intrinsic Parameters

TABLE B.1. Calibrated Parameters

Parameters	Values	Remarks
a_x	274.2	$a_x = f/d_x$
a_y	269.8	$a_y = f/d_y$
u_0	159.7	Ideal optical center=160
v_0	124.0	Ideal optical center=120

C. Fuzzy Rules

No.	Rules
1	IF deviation is extremely left AND distance is near THEN the speed is brake AND steering is steer centre
2	IF deviation is extremely left AND distance is ideal THEN the speed is brake AND steering is steer extreme left
3	IF deviation is extremely left AND distance is far THEN the speed is normal AND steering is steer left
4	IF deviation is left AND distance is near THEN the speed is brake AND steering is steer centre
5	IF deviation is left AND distance is ideal THEN the speed is normal AND steering is steer left
6	IF deviation is left AND distance is far THEN the speed is accelerate AND steering is steer left
7	IF deviation is centre AND distance is near THEN the speed is brake AND steering is steer centre
8	IF deviation is centre AND distance is ideal THEN the speed is normal AND steering is steer centre
9	IF deviation is centre AND distance is far THEN the speed is accelerate AND steering is steer centre
10	IF deviation is right AND distance is near THEN the speed is brake AND steering is steer centre
11	IF deviation is right AND distance is ideal THEN the speed is normal AND steering is steer right
12	IF deviation is right AND distance is far THEN the speed is accelerate AND steering is steer right
13	IF deviation is extremely right AND distance is near THEN the speed is brake AND steering is steer centre
14	IF deviation is extremely right AND distance is ideal THEN the speed is brake AND steering is steer extreme right
15	IF deviation is extremely right AND distance is far THEN the speed is normal AND steering is steer right
16	IF deviation is somewhat centre AND distance is far THEN the speed is accelerate AND steering is steer centre
17	IF deviation is slightly left AND distance is far THEN the speed is normal AND steering is steer centre
18	IF deviation is slightly right AND distance is far THEN the speed is normal AND steering is steer centre

D. Final Prototype (FV)

Acknowledgements

This research was supported by Multi-Agent System for Data Link Interaction in SPALAN Grant (grant number: 1001/PELECT/814265).

References

[1] A. Pfordten, "Lembaga Lebuhraya Malaysia (LLM) Caretaker of The Highways," Jurutera, vol. 2014, pp. 9–11, July 2014.

[2] MIROS, "General Road Accident Data in Malaysia (1997 – 2016),"2016.[Online]. Available: https://www.miros.gov.my/1/page.php?id=17

[3] Bernama, "5,083 Road Accident Deaths Recorded from Jan-Sept 2017," New Straits Times, Nov 2017. [Online]. Available:https://www.nst.com.my/news/nation/2017/11/302157/5083-road-accident-deaths-recorded-jan-sept-2017.

[4] V. Babulal, "Number of Fatal Road Accidents up in 2016, More than 7,000 Lives Lost," New Straits Times, Jan 2017.[Online]. Available: https://www.nst.com.my/news/2017/01/205090/number-fatal-road-accidents-2016-more-7000-lives-lost

[5] P. S. Mooi, "Asian Highway Development Issue," The Ingenieur, pp. 20–28, April 2016.

[6] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision based Intelligent vehicles: State of the Art and Perspectives," Robotics and Autonomous Systems, vol. 32, no. 1, pp. 1-16, 2000.

[7] C. Jiangwei, J. Lisheng, G. Lie, W. Rongben, and L. Bibing, "Study on Method of Detecting Preceding Vehicle Based on Monocular Camera," in 2004 IEEE Intelligent Vehicles Symposium. IEEE, 2004, pp. 750–755.

[8] W. Zheng. (2015) Self Driving RC Car.[Online]. Available:https://zhengludwig.wordpress.com/projects/self-driving-rc-car/

[9] G. Bradski and A. Kaehler, Learning OpenCV, 1st ed.O'Reilly, 2008.

[10] I. Saleh and W. Rahiman, "The Effect of Illumination Compensation Methods with Histogram Back Projection for Camshift Application," Journal of Fundamental and Applied Sciences, vol. 9, no. 6S, pp. 618–631, 2017.

[11] E. Rich, K. Knight, and S. B. Nair, "Heuristic Search Techniques," in Artificial Intelligence, 3rd ed. Tata McGraw Hill Education Pvt. Ltd., 2010, ch. 3, pp.52–76.

[12] P. Verkhogliad. (2012) Fuzzy Logic Example #2. [Online]. Available:https://petro.tanrei.ca/fuzzylogic/fuzzy_meehaan_joy.html1

[13] G. Meehan and M. Joy, "Animated Fuzzy Logic," Journal of Functional Programming, vol. 8, no. 05, pp. 503–525, 1998.

[14] M. Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems, 3rd ed. Pearson Education, 2011.