SIMULATION OF LANDMARK APPROACH FOR WALL FOLLOWING ALGORITHM ON FIRE-FIGHTING ROBOT USING V-REP

Sumarsih Condroayu Purbarani, Qurrotin A'yunina, and Muhammad Anwar Ma'sum

Faculty of Computer Science, Universitas Indonesia, Kampus UI, Depok, 16424, Indonesia

E-mail: muhammad_anwar@cs.ui.ac.id

Abstract

Autonomous mobile robot has been used in almost all sectors in human world, such as in human safety where these kinds of robots pay a lot of contributions. One example of these safety autonomous mobile robots is the fire-fighting robots, which is our main focus in this article. As an autonomous robot, fire-fighting robot needs a robust navigation ability to execute a given task in the shortest time interval. Wall-following algorithm is one of several navigating algorithm that simplifies this autonomous navigation problem. As a contribution, we propose two combinations of method that can be used to make the existing wall-following algorithm more robust. This is done by comparing the combined wall-following algorithms with the original. By doing so, we could determine which alternative has more impact on the robot's navigation robustness. Our goal is to see which method is more effecttive when combined with the wall-following algorithm.

Keywords: *wall-following, landmark, autonomous, navigation, fire-fighting*

Abstrak

Autonomous mobile robot telah banyak diimplementasikan di hampir semua sektor kehidupan manusia, termasuk di sektor penyediaan keselamatan yang juga membutuhkan kontribusi dari *autonomous mobile robot* ini. Sebagai contoh, robot pemadam kebakaran yang juga menjadi fokus utama pembahasan pada artikel ini. Sebagai robot *autonomous,* robot pemadam kebakaran memerlukan kemampuan navigasi yang baik untuk dapat melakukan tugas yang diberikan dalam kurun waktu sesingkat mungkin. Algoritma *wall-following* merupakan salah satu algoritma navigasi yang mampu menyederhanakan permasalahan navigasi ini. Sebagai kontribusi, kami mengajukan dua metode kombinasi yang dapat meningkatkan kemampuan algoritma *wall-following* yang sudah ada. Dengan membandingkan algoritma *wall-following* hasil kombinasi dengan algoritma asalnya, didapat hasil yang menunjukkan metode alternatif mana yang memiliki dampak lebih besar pada peningkatan kemampuan navigasi robot. Tujuan eksperimen ini ialah melihat metode manakah yang lebih efektif untuk dikombinasikan dengan algoritma *wall-following.*

Kata Kunci: *wall-following, landmark, autonomous, navigation, fire-fighting*

1. Introduction

Nowadays, the trend of autonomous mobile robot development is significantly increasing. The development of this type of robot has minimized human's intervention in many ways. Sojourner that is used during the Pathfinder mission to explore Mars is just one of the powerful autonomous mobile robots that have been developed. Aibo, a cute little puppy robot which is manufactured to be a pet, also an autonomous mobile robot that is used for elderly care as one of the research output of University of Kaiserslautern, Germany [1] have shown their ability to do the commanded task autonomously or without human intervention.

Indeed, autonomous mobile robot has been used in almost all sectors in human world, namely industrial, academic, military, etc. In military use we know Predator UAV (Unmanned Aerial Vehicle), a robot which is used to carry missiles. In industrial sectors, numerous mobile robots have been developed to do various precision-demanded tasks. Moreover, in safety providing, kinds of autonomous mobile robot also pay contributions, for instance fire-fighting robots which is our main focus in this article.

Fire-fighting robot which is operated autonomously does its navigation also autonomously. Thus it needs to improve its navigation ability to do the given task in the shortest range of time. Wall-following algorithm is only one of several navigating algorithm that simplifies this autonomous navigation problem.

Wall-following algorithm is proposed to provide a vertical-surface tracking [2]. Using sensors, robot may follow the wall to navigate itself. In

this work, we use a two-wheel and two-motor mobile robot that is an ideal robot for research use, called Pioneer P3DX robot. By default, it comes with 16 ultrasonic sensors and The Braitenberg Algorithm to simply avoid the obstacles i.e. the wall. Ultrasonic sensors are used to follow the wall on its right side then after completing one cycle, follow the wall on its left side. As a contribution we propose two combinations of methods that can be used to robust the existing wall-following algorithm. By comparing these combined wall-following algorithms with the original one, we get the result of which alternative has more impact on the robot's navigation robustness. Our goal is to see which method is more effective when combined with wall-following algorithm.

Related Works

Wall-following algorithm has been widely implemented in autonomous mobile robotic field. Some of those implementations are demonstrated by [2] where the algorithm is implemented using laser scanner to exhibit the efficacy of robot's controller for various initial states along with the experiment in particular distinct environments. [3] also presents the wall-following algorithm using laser range finder to detect the corners and edges. [4] and [5] represents the wall as a set of objects and uses infrared sensor to support the wall-following algorithm. Furthermore, using a hexapod robot, [6] implementing a data-driven fuzzy controller to control the wall-following algorithm. Nonetheless, to solve some classical wall-following problems, [7] classifies task using an associative approach. Overall works state that the wall-following algorithm is an efficient way to reach the target points that link to the wall [8].

2. Methods

Robot's Kinematic

Robot's position on a planar medium is represented by three components, i.e. two Cartesian coordinates and one heading direction. Those components are symbolized as x, y, and θ respectively. To define the robot position, the reference point should be determined. There are two reference frames namely global and local reference frame to be considered in order to determine the robot's position. Global reference is simply robot's position on the Cartesian Coordinates of the real world. The local reference is robot's position according to its own Cartesian Coordinates (see Figure 1).

As we see from Figure I, (X_G, Y_G) is our global reference frame while (X_L, Y_l) is our local reference frame. Since we do calculation only in one
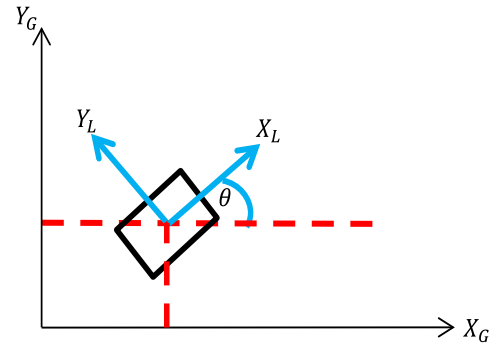

Figure 1. Robot's position on global and local reference frames.

of the frames, we have to convert whether the global reference to local reference or vice versa. The relationship between the frames can be shown by equation(1) below.

$$\xi_L = \mathcal{R}(\theta)\xi_G \tag{1}$$

where ξ_L represent the robot's position on global frame (X_L, Y_l) while ξ_G represent the robot's position on local frame (X_G, Y_G). $\mathcal{R}(\theta)$ is the operation matrix to map ξ_G on the local frame ξ_L and can be shown as in equation(2). As mentioned above, ξ_G is a position, thus it should contain the three-component of robot position. It can be written as the following equation(3).

$$\mathcal{R}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\xi_L = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3}$$

Now that we map the robot position into local frame, we can do the calculation of robot's kinematic constraint that involves robot local reference frame as shown in Figure 2.

The Pioneer P3DX robot has two standard wheels as its actuators. To make it move, we should consider the wheel constraints in order to meet the most proportional velocity. These constraints are sliding constraint and rolling constraint [9].

To reach the stability robot, the velocities of both constraints have to be maintained. The sum of rotational and translational velocity of sliding constraint has to be zero. It means that the wheel does not slide. This procedures can be seen in equation(4) and (5).
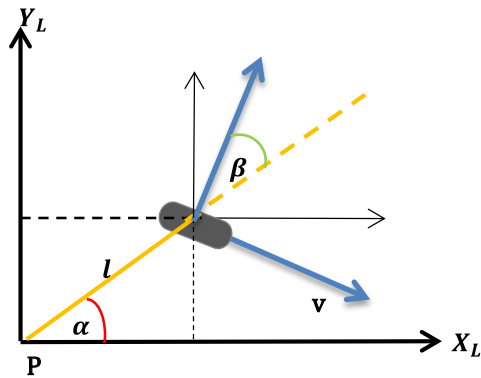
$$v_{trans} + v_{rot} = 0 \tag{4}$$

Figure 3. Robot's kinematic components in local reference frame.

$$\begin{bmatrix} cos(\alpha + \beta) \\ sin(\alpha + \beta) \\ l\,sin\,\beta \end{bmatrix}^{T} \mathcal{R}(\theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0 \qquad (5)$$

where, l is radius between the center point of robot chassis and the observed wheel, α angle between radius l and X coordinate of robot's local reference, β = angle between the extension of radius l and the perpendicular resultant of wheel velocity, $\dot{x}$ = wheel's velocity on x coordinate in the global reference, $\dot{y}$ = wheel's velocity on y coordinate in the global reference, and $\dot{\theta}$ wheel's velocity of the angle θ of heading direction in the global reference.

On the other hand, the component of rolling velocity has to equals to the velocity of the wheel. It means that the sum of translational and rotational velocity of the rolling constraint equals to the wheel velocity, as written in equation(6) to equation(9).

$$v_{rolling} = v_{wheel} \qquad (6)$$

$$v_{rolling} - v_{wheel} = 0 \qquad (7)$$

$$v_{trans} + v_{rot} - v_{wheel} = 0 \qquad (8)$$

$$\begin{bmatrix} sin(\alpha + \beta) \\ -cos(\alpha + \beta) \\ -l\,sin\,\beta \end{bmatrix}^{T} \mathcal{R}(\theta) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} - r\dot{\varphi} = 0 \qquad (9)$$

where r and $\dot{\varphi}$ represents wheel's radius and wheel's spinning velocity

The Algorithms

In this work, we compare the two different approaches of landmark implementation and see which approach is best supports the wall-following algo-
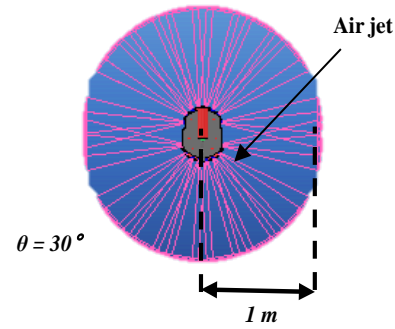

Figure 2. Ultrasonic sensor on Pioneer P3DX robot.

rithm. The robot is given a task to find the fire source and once the robot finds it, robot has to extinguish it. An air jet is installed on the robot's top part to extinguish the fire. The minimum of distance from the wall is defined as the initial point. It is manually defined by user in order to avoid the crash with the wall.

We use all the 16 ultrasonic sensors (6 in the front, 2 on the right, 2 on the left, and 6 on the back) the Pioneer P3DX robot is equipped with. For the sensor volume we choose cone with the angle of 30°. The range of the sensor is set to 1 m (see Figure 2).

According to the Wall-following Algorithm, robot follows the wall whether the one on its left or right. In this work, the initial direction to follow is set to right by default. The minimum of distance from the wall is defined as the initial point.

The Braitenberg Algorithm is included in order to avoid collision with any obstacle near the robot. In this case its only obstacle is the wall. The Braitenberg Algorithm simply makes use of sensor information to adjust the robot's position. The ultrasonic sensor scans continuously to the wall and return the approximate value of the robot's position from the wall. Using the information from the sensors' observation result, robot has knowledge to make a decision whether to turn to keep on moving forward or to turn over.

The Two Approaches

As mentioned before, two different methods are used to robust the robot's navigation using wall-following algorithm. Both methods are actually based on a logic i.e. making landmark by coloring. The colors in both algorithms are assigned as a checkpoint for the robot to change its direction. For this reason we add some vision sensor to detect the color appearance and also to see if there is a fire or not. The difference between the methods is the medium where the color is implemented to and the algorithm approach of each method.

The Wall-following Algorithm with Horizontal Landmark marks red and blue colors to a horizontal surface, in this case is to the floor as shown in Figure 4.This approach uses 2 vision sensors one of which is attached in the front part and another one is attached at the bottom part of the robot.

The Wall-following Algorithm with Vertical Landmark uses green and blue colors to be marked on the wall as shown in the Figure 5. It uses 3 vision sensors that are installed on the right, left and in the front part of the robot. The flowchart of both methods' algorithm is shown in the Figure 6 to make it clearer.

First of all, the ultrasonic sensors detect robot's surrounding, check whether there is wall or not by measuring the robot's distance from the 'obstacle'. By default, the robot is set to follow the right wall at first. Thus, once the sensors on the robot's right side detect the presence of wall on the right side it will start following the right wall. It will maintain its distance from the wall in a certain constant value continuously until the next command. If the front ultrasonic sensors detect an 'obstacle' which means that it reach a corner, then robot will decide to turn to the greater distance, in this case, to the left.

If the robot's vision sensor sees the fire, or the fire appears in the camera, then robot will stop on a certain distance from the fire. It will extinguish it by activating the air jet. It will also terminate the robot's task.
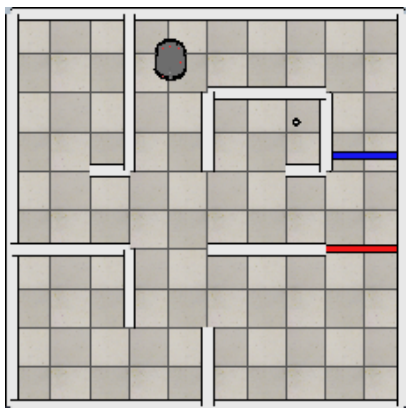

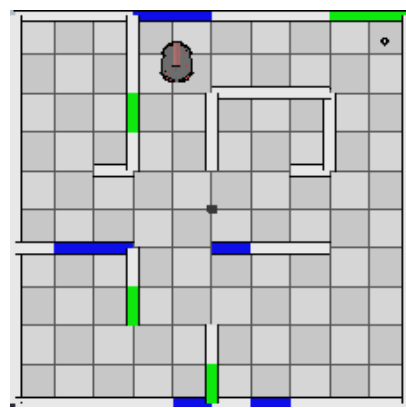Figure 6. The appearance of the Wall-following Algorithm with Horizontal Landmark.

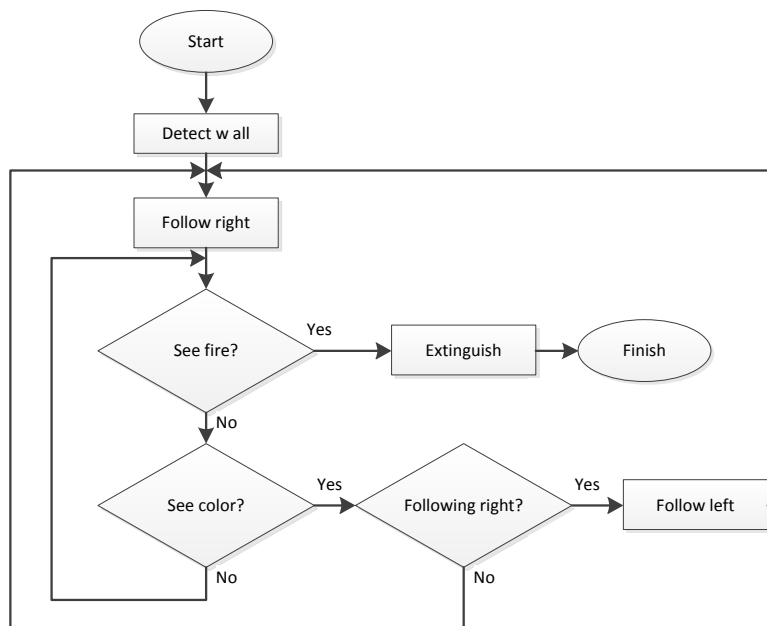Figure 5. The appearance of the Wall-following Algorithm with Vertical Landmark.

Figure 4. The Wall-following Algorithm with Landmark.

TABLE 1
TIME ELAPSED AND SUCCESSION RATIO FOR HORIZONTAL LANDMARK

Velocity (m/s)	S1 (s)	S2 (s)	S3 (s)	S4 (s)	S5 (s)	S6 (s)	S7 (s)	Total Time (s)	Succession Rate (time)	Succession Ratio (%)	Average Time
Horizontal Landmark - Map 1											
v1	F	F	F	F	F	F	F	0	0	0	0
v2	257	34	F	99	92	143	F	625	5	71	125
v3	177	F	14	67	60	92	F	410	5	71	82
Horizontal Landmark - Map 2											
v1	F	F	F	F	F	F	F	0	0	0	0
v2	256	34	F	104	96	183	F	673	5	71	135
v3	175	F	15	67	61	128	F	446	5	71	89
Horizontal Landmark - Map 3											
v1	F	F	F	F	F	F	F	0	0	0	0
v2	229	34	F	104	96	185	F	648	5	71	130
v3	158	F	13	66	60	127	F	424	5	71	85

Where:
v   = velocity
S   = position of the fire source
F   = failed

TABLE 2
TIME ELAPSED AND SUCCESSION RATIO FOR VERTICAL LANDMARK

Velocity (m/s)	S1 (s)	S2 (s)	S3 (s)	S4 (s)	S5 (s)	S6 (s)	S7 (s)	Total Time (s)	Succession Rate (time)	Succession Ratio (%)	Average Time
Vertical Landmark - Map 1											
v1	F	47	F	139	145	F	F	331	3	43	110
v2	190	26	F	162	F	F	F	378	3	43	126
v3	F	28	34	58	78	F	F	198	4	57	50
Vertical Landmark - Map 2											
v1	F	F	F	114	116	F	F	230	2	29	115
v2	108	F	F	27	53	F	F	188	3	43	63
v3	F	F	F	F	F	F	F	0	0	0	0
Vertical Landmark - Map 3											
v1	F	F	F	53	78	F	F	131	2	29	66
v2	130	F	F	60	61	F	93	344	4	57	86
v3	F	F	F	F	F	F	F	0	0	0	0

Where:
v   = velocity
S   = position of the fire source
F   = failed

If robot's vision sensors detect any landmark color then robot knows it has to change its direction from the previous one. If it were following the right wall, now it starts following the left wall. It will maintain its distance from the left wall in a certain constant value.

3.   Results and Analysis

We use V-Rep robot simulator as the environment to test the robot. The installation of the environment is the same. We use three different maps to be test to each method. These maps can be found in the Appendix 1. To test in which velocity the robot can perform its best performance in accomplishing the given task. We give three-velocity test to each robot, namely v1, v2 and v3 with value of 1 ms^{-1}, 2 ms^{-1} and 3 ms^{-1} respectively for each map.

We want to compare the succession ratio and elapsed time of each experiment. Succession ratio is a percentage of the total reward for each map.

The reward is given to the robot each time the robot finds the fire source. 100% of succession ratio means that the robot has accomplished to find all of possible position of fire sources in one map. In addition we also collect the average of elapsed time needed to find all fire sources in one map to see which approach is faster than the other. The result is shown in Table 1 and Table 2.

From the above result we can conclude the succession ratio and the elapsed time between the two approaches on each map into graphs as shown in Figure 7, Figure 8, and Figure 9.

As seen in Figure 7, the Horizontal Approach dominates the succession ratio on the Map 1 when controlled by velocity of v2 and v3. This approach completely fails when run with v1 as shown in Figure 11. Robot stuck in a certain position as shown in the Figure 10 where neither x-axis nor y-axis changes after a certain time. This shows that with the defined minimum distance from the wall to run this approach the minimum velocity should be greater than 1 ms^{-1}. According
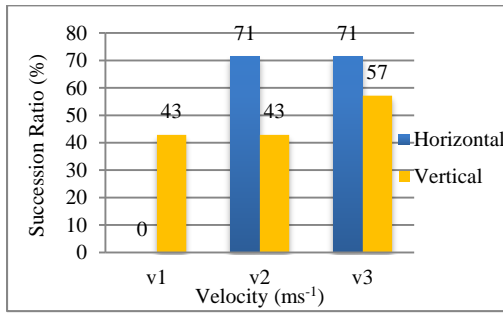
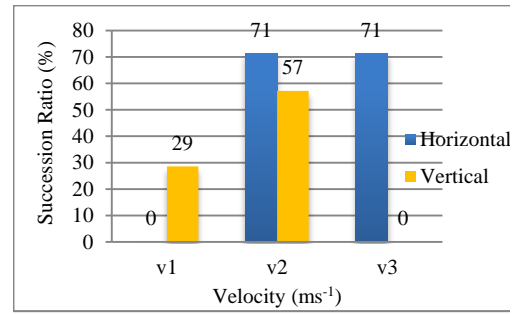Figure 10. Succession ratio of the Two Approaches – Map 1.

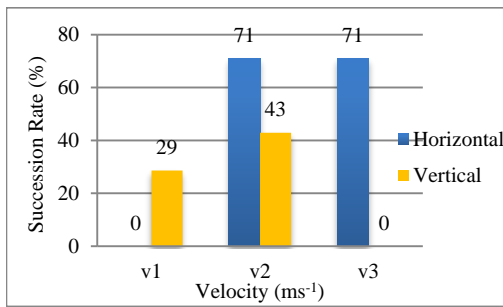Figure 8. Succession ratio of the Two Approaches – Map 3.

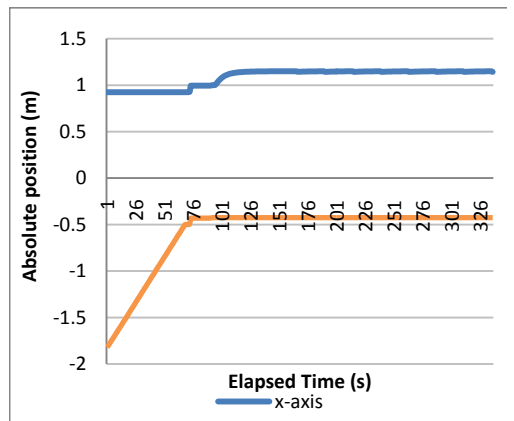Figure 9. Succession ratio of the Two Approaches – Map 2.

Figure 7. XY position – time graph related to Figure 11.

to the data in the Table 1, robot also fails to find the fire source at S3 and S7 with the velocity of v2 as well as at S2 with the velocity of v3.

The Vertical Approach shows an intermediate performance on the Map 1 with four failures when run with the velocity of v1 and v2, and three failures when run with v3. This shows us that v1, v2, and v3 are actually still in the range of available velocities required by the robot when run this approach. For the case of S1, S6, and S7 in all velocities (except S1 in v2), they are unreachable by robot because robot keeps on wandering around a certain region on the way to the mentioned goals. It is as represented by the red rectangle shown in Figure 13. Yet, somehow S1 in v2 can be reached. This is due to uncertainty of the sensor reading that inevitably happen in robotic world.

On the Map 2 (see Figure 8), the Horizontal Approach shows a similar performance as the one on the Map 1. The failures happen when robot tries to get to S3 and S7 with the velocity of v2 also S2 and S7 with the velocity of v3 (see Figure 12). These happen because robot experiences a blind spot that robot's camera detects no object even though it is just passing the fire as shown in the red rectangle part of Figure 12.

The succession ratio of the Vertical Approach on the Map 2 is not as good as the previous map. Here, this approach totally fails to find any fire source using velocity of v3. The change in the

map configuration also changes the position of the landmark. Robot again keeps on wandering around a certain region as represented by the red rectangle shown in Figure 13. This also shows that this approach has lack in flexibility that leads to invalidity.

For the third map, Map 3, the Horizontal Approach again shows the same ratio of success-ion. The unreachable fire sources are also the same as previous experiment.

The succession ratio of the Vertical Approach on the Map 3 does not show any difference from the previous experiment when it is run with v1 and v3. Yet it shows a bit improvement by reaching the goal on S7 with v2 which none of the experiments can make. See Appendix 2 for an example of a success-fully completed task of Horizontal Landmark (v2, S1, Map 1) and Appendix 3 for an example of a successfully completed task of Vertical Landmark (v2, S1, Map 1) frame by frame.

According to the explanation above we can summarize it by giving a reward point (P) for the better succession ratio than the other on each map as the Table 3. Thus, we can conclude that Horizontal Landmark get the highest succession ratio regardless the elapsed time.
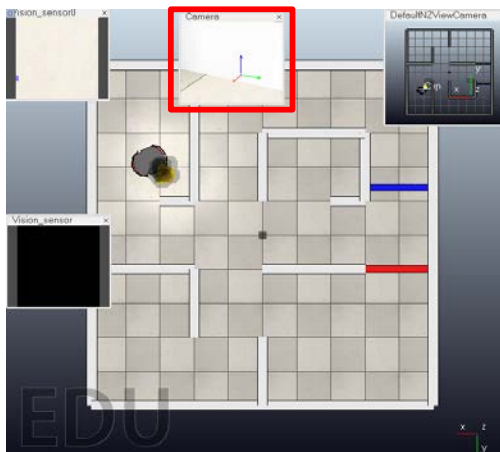
Figure 12. Robot get stuck in a certain position.

Figure 13. Robot keeps on wandering around a certain region.

TABLE 3
SUCCESSION RATE COMPARISON RESULT

	v1	P	v2	P	v3	P	Total
HL - Map 1	0%	0	71%	1	71%	1	2
VL - Map 1	43%	1	43%	0	57%	0	1
HL - Map 2	0%	0	71%	1	71%	1	2
VL - Map 2	29%	1	43%	0	0%	0	1
HL - Map 3	0%	0	71%	1	71%	1	2
VL - Map 3	29%	1	57%	0	0%	0	1
					Horizontal Landmark =		6
					Vertical Landmark =		3

Note:
HL - Map 1 represents Horizontal Landmark – Map 1, VL- Map 1 represents Vertical Landmark – Map 1, and so on.

Figure 11. Robot is experiencing a blind spot.

TABLE 4
ELAPSED TIME COMPARISON RESULT

	S1	S2	S3	S4	S5	S6	S7	Total
HL - Map 1	1	0	1	1	2	2	0	7
VL- Map 1	1	3	0	2	1	0	0	7
HL - Map 2	1	1	1	1	1	2	0	7
VL- Map 2	1	0	0	2	2	0	0	5
HL - Map 3	1	1	1	1	1	2	0	7
VL- Map 3	1	0	0	2	2	0	1	6
					Horizontal Landmark =			21
					Vertical Landmark =			16

Now that we know the best succession ratio on each map, we have to look at the elapsed time to support the validity of the former succession ratio. We compare the elapsed time that each approach needs to find each of the fire sources with various velocities. As we done before we give a reward point (P) equals to 1 to each case with smaller elapsed time as shown in Table 4. The values on the cells are the sums of reward point of all velocities to get to one fire source. For more details, see Appendix Table 1

As we can see from the Table 4, when we compare the elapsed time of each velocity on each map, we get the result that by using the Horizontal Landmark, robot can increase the time efficiency. It at least presents a better alternative for time efficiency than the other algorithm. Combined with the succession ratio, finally we can conduct the final result that the more robust approach in wall-following algorithm is provided by the Horizontal Landmark.
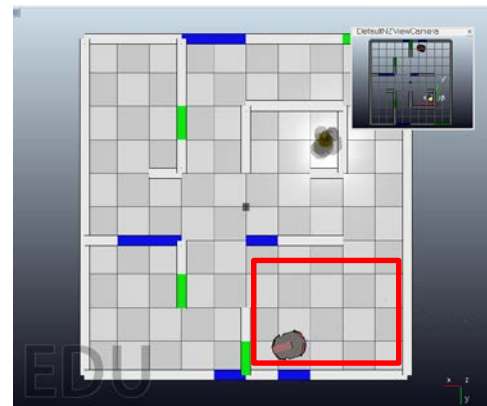
4. Conclusion

To make a robot move we have to consider the minimum velocity of it to get the most stable motion. If the velocity is less than the minimum requirement, robot won't be able to perform the maneuvers to avoid obstacles. On the other hand robot may also experience instability during motion if the velocity is much greater than required. Thus, the robot also has its maximum velocity that should not be exceeded.

Robot may also fail to find some fire sources while succeeds to find the others even though the velocity is already set to be greater than the minimum velocity. It is caused by the minimum distance of robot position from the wall. It may cause the robot to have blind spots when it is further

than the distance between the fire source and the wall.

Uncertainty is another source of error in the experiment. Even though we use a robotic simulator, the uncertainty is inevitable. This uncertainty may arise from environment, sensors, actuators, the algorithm, error that occurs during the modeling on the simulator itself, and any other unpredictable noises [10]. The environment may subject to error due to the wall configuration and the landmark positioning both on the floor or wall. The sensors pay contribution to such error as well. Error may occur during observation that sensors present measurement data to far from accuracy. To alleviate this error, equipping the robot with more sensors may be a solution. Albeit it can extinguish all error, it can at least minimize the error caused by the sensors. Another source of error is actuator as the user of the sensor data.

The inability of the Vertical Landmark approach to prove its robustness compared to the other one is mostly caused by the inflexibility of the landmark when it is implemented in different environment.

At last, the Horizontal Landmark method can be used as an alternative approach to be implemented with wall-following algorithm. This approach is still modifiable and needs lots of improvements to robust the fire-fighting robot navigation on a planar environment even more. Our future work will be based on this approach.

Acknowledgement

References

[1] Berns, K.; Mehdi, S. A., "Use of an autonomous mobile robot for elderly care," *in Advanced Technologies for Enhancing Quality of Life (AT-EQUAL)*, Romania, 2010.

[2] Karl Bayer; Avik De; D. E. Koditschek, "Wall Following for Autonomous Navigation," *in SUNFEST, University of Pennsylvania*, Summer 2012, Pennsylvania, 2012.

[3] Nepali, M. R.; Yadav, N.; Prasad, D. A. H.; Balasubramaniam, S.; Venkatesh EN; Ashutosh, "A Novel Wall Following Algorithm For Mobile Robots," *International Journal of Robotics and Automation (IJRA),* vol. 5, no. 2, pp. 15-23, 2014.

[4] P. Van Turennout, G. Honder d, L. J. van Schelven, "Wall-following control of a Mobile Robot," *in IEEE, International Conference on Robotics and Automation*, 1992.

[5] J. Borenstein, Y. Kor en, "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots," *IEEE Transaction on Robotics and Automation*, vol. 7, no. 3, pp. 278-288, June 1991.

[6] Juang, C.F.; Chen, Y.H.; Jhan, Y.H., "Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 611-619, 2015.

[7] Navarro, Rodolfo; Acevedo, Elena; Acevedo, Antonio; Martinez, Fabiola, "Associative model for solving the wall-following problem," *in Pattern Recognition, J. C. e. al., Ed., Berlin, Springer*, 2012, pp. 176-186.

[8] M. Swati and B. Pankaj, "Maze solving algorithms for micro mouse," *in IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2008. SITIS'08., 2008.

[9] R. Siegwart and I. R. Nourbakhsh, Introduction to autonomous mobile robots, London: MIT press, 2004.

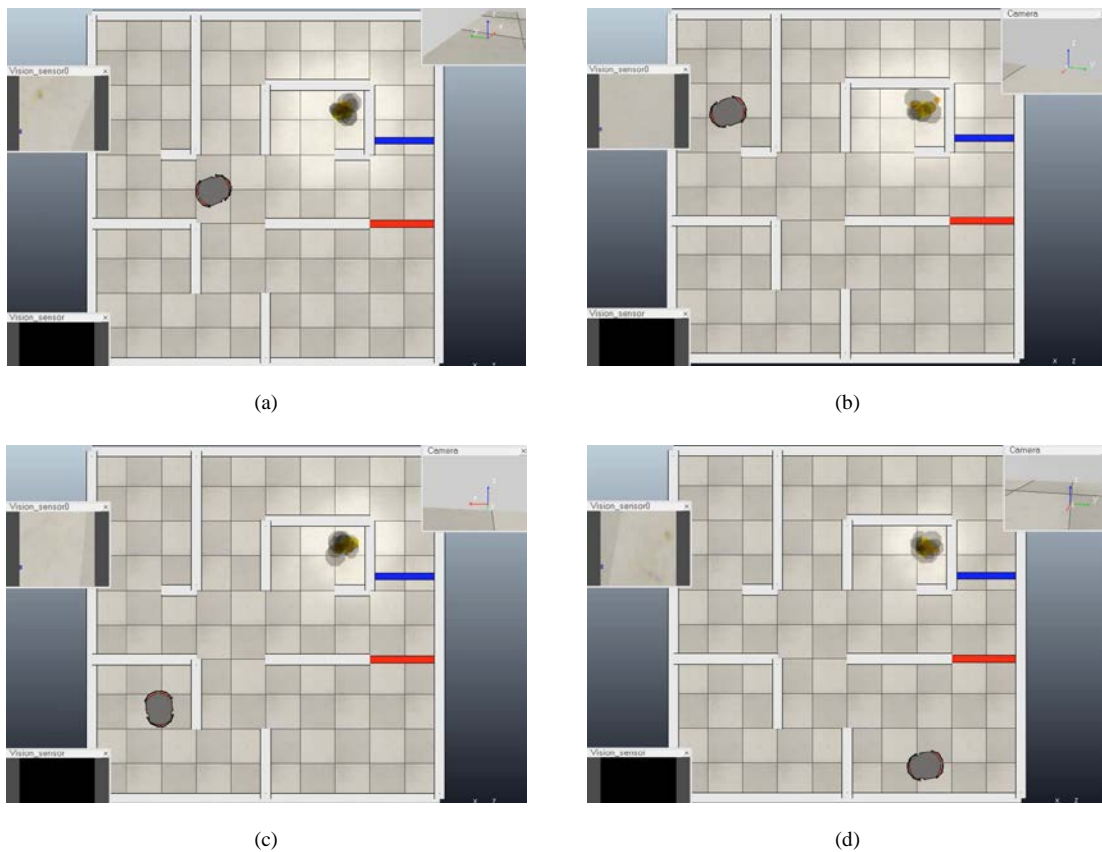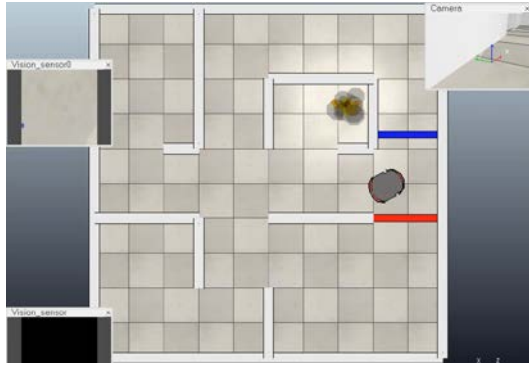[10] Sebastian Thrun, Dieter Fox, Wolfram Burgard, Probabilistic Robotics, 1999.

Appendix

Maps

Map 1 to 3 is shown in the figures below. The red dots represent the positions of the fire sources.

<p align="center">(a)</p>

<p align="center">(b)</p>

<p align="center">(c)</p>
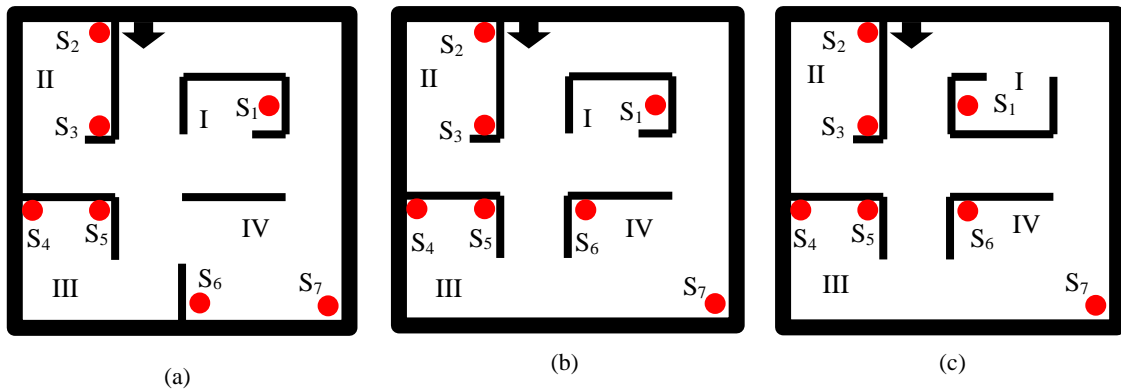
Appendix 1. (a) Map 1; (b) Map 2; (c) Map 3; I, II, III, IV represents the region

<p align="center">(a)</p>

<p align="center">(b)</p>

<p align="center">(c)</p>
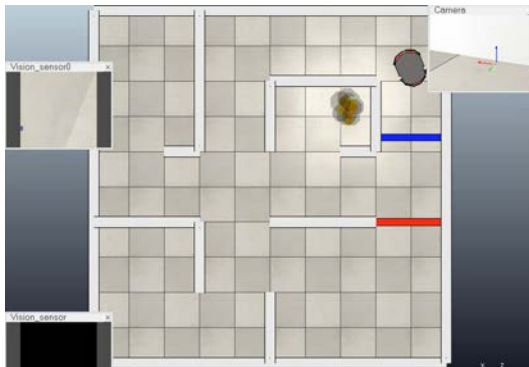
<p align="center">(d)</p>
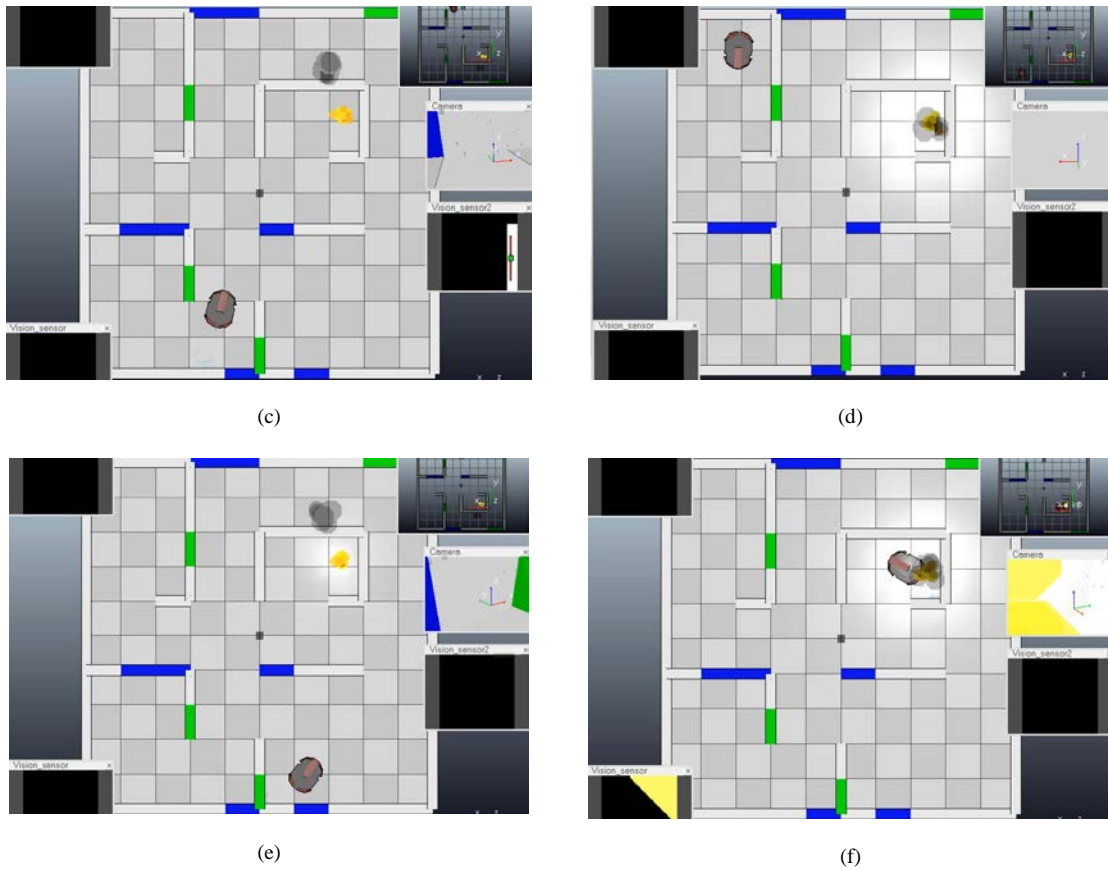
(e)

(f)

(g)

(h)

Appendix 2. Frames from one of successfully completed task of Horizontal Landmark Approach. (a) Robot starts to observe the Region II; (b) Robot is observing the Region II; (c) Robot observes the Region III; (d) Robot observes the Region IV; (e) Robot change the direction after passing the red landmark, thus, (f) it comes back to the Region IV; (g) this time when passes the red landmark, robot changes the direction again hence it does not came back observing the Region IV; (h) Robot finds the fire source in the Region I.

(a)

(b)

(c)

(d)

(e)

(f)

Appendix 3. Frames from one of successfully completed task of Vertical Landmark Approach. (a) Robot makes its first move; (2) Robot observes the Region III at first; (c) Robot exits from Region III and then starts to (d) observe the Region II; (e) Robot observes the Region Iv; (f) Robot finds the fire source in the Region I.

APPENDIX TABLE 2
ELAPSED TIME COMPARISON RESULT (DETAILS)

	S1			R	S2			R	S3			R	S4			R	S5			R	S6			R	S7			R	Ttl
	v1	v2	v3		v1	v2	v3		v1	v2	v3		v1	v2	v3		v1	v2	v3		v1	v2	v3		v1	v2	v3		
HL1	-	257	177	1	-	34	-	0	-	-	14	1	-	99	67	1	-	92	60	2	-	143	92	2	-	-	-	0	7
VL1	-	190	-	1	47	26	28	3	-	-	34	0	139	162	58	2	145	-	78	1	-	-	-	0	-	-	-	0	7
HL2	-	256	175	1	-	34	-	1	-	-	15	1	-	104	67	1	-	96	61	1	-	183	128	2	-	-	-	0	7
VL2	-	108	-	1	-	-	-	0	-	-	-	0	114	27	-	2	116	53	-	2	-	-	-	0	-	-	-	0	5
HL3	-	229	158	1	-	34	-	1	-	-	13	1	-	104	66	1	-	96	60	1	-	185	127	2	-	-	-	0	7
VL3	-	130	-	1	-	-	-	0	-	-	-	0	53	60	-	2	78	61	-	2	-	-	-	0	-	93	-	1	6

Horizontal Landmark =   21

Vertical Landmark =   18

Note:

HL1 represents Horizontal Landmark – Map 1, VL1 represents Vertical Landmark – Map 1, and so on.