Fuzzy Logic Based Framework for Mobile Robot Navigation with Target Tracking

[1] Tharindu Fernando, [2] Harshala Gammulle, [3] Chamila Walgampaya

[1, 2] Department of Statistics and Computer Science
Faculty of Science, University Of Peradeniya,
Peradeniya, Sri Lanka.

[3] Department of Engineering, Mathematics
Faculty of Engineering, University Of Peradeniya,
Peradeniya, Sri Lanka.

Abstract - With the advances of technology, mobile robots are becoming increasingly popular. According to Karthiga (2014) developed countries get the assistance of mobile robots to rescue humans in disaster areas and this has been proven as a productive method to eliminate human error. In such instances robots are required to navigate in hostile environments such as collapsing buildings or areas affected fire. This paper presents implemented control architecture for mobile robot target tracking and obstacle avoidance in a dynamic hostile environment. Given the existing body of research results in the field of obstacle avoidance and path planning, which is reviewed in this context, particular attention is paid to integrate computer vision based sensing mechanisms to robust fuzzy logic based navigation control method. A rule-based fuzzy controller with reactive behaviour was implemented and tested on a RP5 mobile robot platform. Depth and colour information for both navigation and target tracking are to be captured using a Asus Xtion PRO sensor. This traversability data is used to infer, in real time, the navigational path based on the Fuzzy Rule-Base algorithm. The effectiveness of the proposed approach was verified through several experiments, which demonstrates the feasibility of a fuzzy target tracker as well as the extensible obstacle and hostile region avoidance system.

Keywords - *Fuzzy Logic Control, Mobile Robot Target Tracking, Obstacle Avoidance Module, Hostile Region Avoidance Module.*

1. Introduction

In recent years, there has been a growing realization that mobile robots are one of the key factors affecting the success in the service sector. According to Araiza (2010) the use of mobile robots to substitute human beings in high risk tasks has been multiplied due to the increase of flexibility, reliability, robustness, speed, accuracy, and many other advantages that advances in technology have brought. Mobile robots are becoming popular and used extensively in many fields such as military and nuclear power plants as a method of intervention in hostile environments or in warehouse as a method of transport and inventory inspection. Based on the study by Ronald et. al (1990) in computer integrated manufacturing (CIM) systems automatic guided vehicles (AVG) are replaced with mobile robots, as they employee intelligent behaviour to cope with the open real world environments without the constraints that are imposed in AVGs. In dynamic challenging environment finding a hostile condition such as a collapsing building or an opponent may become a possibility. But traditional industrial robots assume that the state of the world is known with near certainty. Therefore there is a need for a robust, intelligent system which can navigate mobile robots dynamically and capable of graceful motion with increased dynamic stability. According to Marzouqi (2014) hostile region avoidance capability of mobile robots is well tested in covert robotics. It is a relatively new field in which covert navigation abilities are developed for robots to carry out different missions. The robot derives various strategies based on the environment and the hostile observers' locations. Based on those strategies a motion plan is generated which minimizes the risk of being detected. As stated by Kolarow (2012), tracking arbitrary objects or persons in video sequences in real-time, is a key condition for many applications in mobile service robotics and Human-Robot Interaction (HRI). The target may vary from a specific coloured object for industrial service robots to tracking humans for surveillance and rescue robots. In fact mobile robots in car manufacturing material handling systems use special barcodes and markers to track and deliver the materials to necessary work station. The inhomogeneous and variant appearances of the dynamic background along with the changing in illumination make it a computational intensive problem.

2. Literature Survey

In many research findings fuzzy logic techniques are tested for the navigation of mobile robots. A fuzzy control module for mobile robot target tracking is proposed by Rashid (2010) et. al. The authors have only tested the approach for motion control but not for obstacle avoidance. Another fuzzy logic based target tracking system for mobile robot is presented in the study conducted by Li (2004) et. al. The system is comprised of infrared (IR) sensors to detect obstacles and two wheeled robots, the first one as the moving target and the second as the tracker.

As stated by Mohammed (2013), apart from IR sensors, Ultra sonic sensors and laser range finders are the most commonly used sensors in depth sensing mechanisms. Deviated from that, several studies have being conducted using camera information to navigate the robot in a dynamic environment. Various methods have been used and proposed based on the characteristics of the environment being traversed and usually relying on segmentation of the image into distinct areas for sub-processing. These systems classify the image by segmenting the region and identifying the available path area in front of the vehicle, indicating a safe area for navigation.

Li (1994) has proposed a combine approach which uses both ultrasound sensing and vision based perception as sensing methods. To achieve this objective, an array of ultrasonic sensors and a vision system were mounted on a mobile robot. The ultrasonic sensors were used to provide distance information between the robot and obstacles for robot navigation by reactive behaviours, such as avoiding obstacles and following edges, while the vision system provided some sub goals for determining a good motion direction to avoid robot trap in local region. Since perception and decision units in this method were integrated in one module, this strategy has the better real-time response and reliability compared to general approaches described by Mohammed (2013) and Li (2004) in their respective studies. Even with these advantages, the overhead of using both ultra sound and vision based perception is still considerable.

With the introduction of low cost 3D cameras several independent studies have emerged in order to exploit the advantages of this equipment in other applications, ranging from healthcare to robotics. Due to the use of infrared, the sensor is able to create the depth map even at environments with total absence of light.

In the system proposed by Santos (2012) et al. an artificial neural network (ANN) is used to recognize different configurations of the environment, for example, path ahead, left path, right path and intersections. The ANN was trained using data captured by the Kinect sensor in indoor environments. The authors proposed the system for surveillance robots as the system was capable to work even under the dark lighting conditions with the use of IR depth sensor in the Kinect sensor. The method lacks in generalisation to different dynamic environments as the ANN should be explicitly trained to track targets and avoid hostile regions.

Kloss (2009) et. al have proposed a method for adaptive colour processing in object recognition problems with the aid of trained neural network. The method has been tested for mobile robots target tracking problem in the study conducted by Benavidez (2011) et. al. The study uses Microsoft Xbox Kinect sensor as the sensing mechanism, which provides RGB colour and depth imaging data. The authors have used this compact sensor for both obstacle avoidance and target tracking tasks.

The proposed methods in Bajones (2014) and Biswas (2012), which uses a similar RGB-Depth camera set up, lacks in their applicability to unstructured environments. Both systems have the capability to localize the robot in the environment, given a probabilistic occupancy grid of the area for the robot to explore. Hence, a predefined goal is represented in (x,y) coordinate pair. These approaches are ideal for robot application in office environments but hardly applicable for rescue robots and covert robots.

3. Materials and Methods

3.1 Asus Xtion PRO Sensor

The sensor is capable to capture colour and depth information separately. It uses a RGB camera, a monochrome camera and reflective infra-red (IR) camera for sensing. The sensor generates a depth map in real time, where each pixel corresponds to an estimate of the distance between the Xtion sensor and the closest object in the scene at that pixel's location. It is expressed as point cloud data in a coordinate system fixed to the monochrome camera. The sensor can capture depth images in 320x240 resolution at a rate of 30 fps (Asus Xtion Pro sensor 2014). The Asus Xtion Pro sensor was selected because of its slightly larger Field Of View (FOV) (58 H x 45 V vs 57 H x 43 V) compared to that of Microsoft Kinect.

3.2 OpenNI SDK

OpenNI (OpenNI.org 2014) is a free and open source driver that enables accessing the depth and colour matrices of Asus Xtion PRO using C++ platform. The depth map produced from the IR camera and the texture map produced by RGB colour map of the scene is obtained via OpenNI driver. According to Choppin (2013) the maximum depth for automatic object recognition through OpenNI is limited to 4m.

3.3 Modelling Dynamic Hostile Environment

The modelled environment is dynamic and hostile. There can be any number of static and dynamic obstacles in the region which can enter and exit the scene at any time. When localising the desired target region and hostile regions, RGB camera of Asus Xtion PRO sensor is used. The target area is indicated using a red colour laser dot. In order to eliminate the noise from other similar coloured objects in the environment, the gain of the RGB camera is decreased. A colour stream with higher energy is emitted by a laser. Therefore the laser dot will still be visible in the extracted frames. This will ensure that the detected colour patches are only from the laser dots. This approach was experimented with shifts in light intensity and was able to obtain acceptable results. The hostile regions are indicated using blue colour laser beams and the existence of multiple hostile regions is possible. The target for the robot is provided using a red colour laser beam and if no such target is provided the robot enters the wondering mode and will start exploring the environment.

3.4 Fuzzy Logic Control

The proposed navigation system is based upon Mamdani fuzzy inference system. Three such controllers are used to navigate the mobile robot from the initial position to the desired goal. As shown in Figure 1, the initial position of the robot is (75, 52) in Cartesian grid. The robot navigates to the goal in position (625,725)

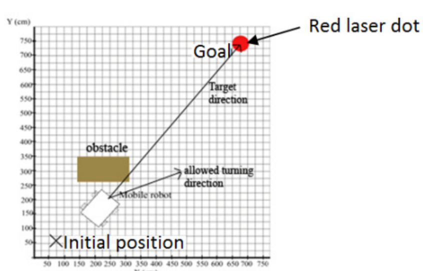

Figure 1: Initial position and the desired target location for the robot.

Figure 2 illustrates the composition of the fuzzy logic based navigation control system. Target Tracking Module (TTM), Obstacle Avoidance Module (OAM) and Hostile Regions Avoidance Module (HRAM) are combined to move the mobile robot to the target along a collision and hostile regions free path. The output of the TTM produces a fuzzy set representing the desired turning direction while the outputs from OAM and HRAM produces fuzzy sets representing the allowed turning directions.

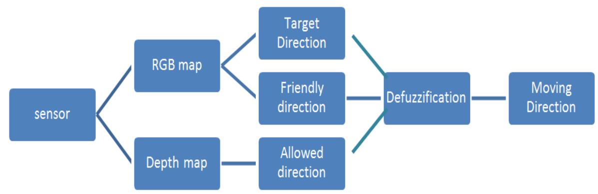

Figure 2: fuzzy logic based navigation

Each control module consists of a set of fuzzy control rules and a fuzzy inference module. Instead of producing an exact value, each control module used produces a fuzzy set. They provide the preferred behaviour of each module. Preferences contain more information, as they give a measure of desirability for each possible command. The control rules could be induced by empirical knowledge. Rule base is composed of many fuzzy implication relations, which are obtained based on lots of experiments, observation and operation experience.

We then use the intersection fuzzy operator, stated by Saffiotti (1997) et. al, to combine the preferences of different behaviours into a collective preference. This command fusion process is graphically illustrated in Figure 3 for the case of two behaviours which both concern about the desired moving direction.

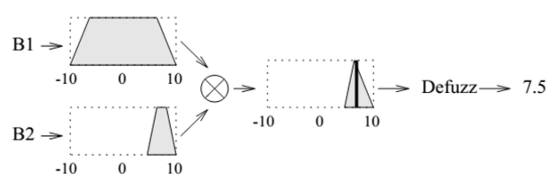

Figure 3: Command fusion with individual preferences. Reference: Saffiotti (1997) et. al

For an example in the above mentioned case the final fuzzy set value *FV(x,y)* will be,

$$FV(x,y) = B_1(x,y) \cap B_2(x,y) \qquad (1)$$

Where x is the input array that defines the actual state and y represents the possible values in the output variable. After combining individual preferences through a command fusion module, the collective preference is defuzzified to produce an exact output value.

The algorithm uses weighted summation defuzzification method (Figure 4). It returns the center of output after aggregating the membership function as the final output. This can be represented as

$$z_{COA} = \frac{\int_0^z \mu_A(z) z dz}{\int_0^z \mu_A(z) dz} \quad (2)$$

where z_{COA} is the exact output, $\mu_A(z)$ is the aggregated membership function and z is the output variable.

According to Naaz (2011) when this set is not unimodal, COA based defuzzication may result in the selection of an undesirable control value, i.e. a value which lays in the gap between two peeks of the combined set, and has low membership in this set. To overcome this problem, when more than one peak exists, the set is divided into a number of separate subsets. The subset which has the largest area is then selected and defuzzified separately using the COA method.

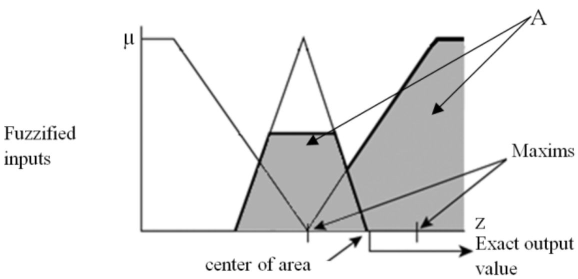

Figure 4: COA defuzzification method

The structure of each module is further elaborated in below subsections.

3.4.1 Target Tracking Module (TTM)

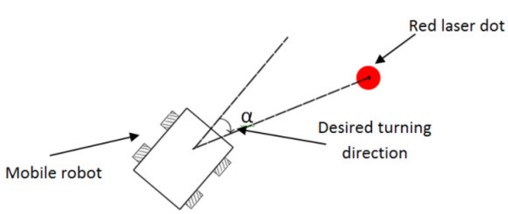

Figure 5: Target direction

As shown in Figure 5 the target to the mobile robot is indicated using a red laser beam. The angle α between the red laser beam and the mobile robot is fuzzified as the desired turning direction.

Hue, Saturation and Value (HSV) colour space separates the intensity information from colour information. Therefore it is the most appropriate colour space for colour based image segmentation where there can be fluctuation in illumination conditions. After the conversion of the RGB colour map produced by the colour camera of the Xtion PRO sensor to HSV stream, it is threshold for red colour to obtain the red colour area in the image. This produces a binary image with intensity value equal to 1 for red colour region and zero otherwise. The optimal thresholding value is determined experimentally using colour histograms, working with different background colours and target positions. As we use only the colour information discarding the intensity information, by converting the RGB stream to HSV stream, the system is less prone to illumination variations. Figure 6 illustrates our approach.

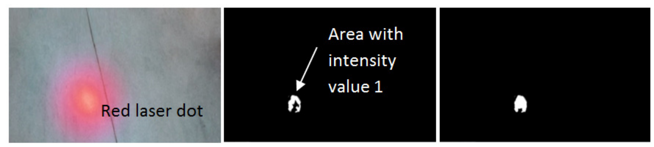

Figure 6 (a): Original frame
Figure 6 (b): threshold frame
Figure 6 (c): Frame after filling holes

The resultant binary image may be corrupted with some degree of noise due to the inconsistencies in the colour stream. Post processing morphological operations such as filling holes, is implemented using OpenCV (OpenCV.org 2014) library. This removes small areas with intensity value equal to zero within a connected component with intensity value equal to 1 (Figure 6 (c)).

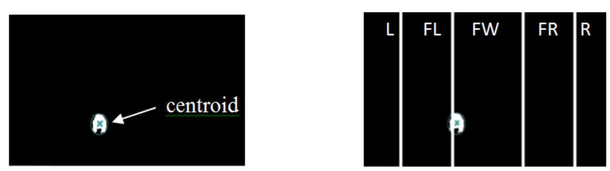

Figure 7 (a): Centroid selection
Figure 7 (b): Frame after partitioning

For each connected region represented with intensity value 1, a centre pixel is approximated by assuming circularity of the regions (Figure 7 (a)). Obtained centroids are superimposed with the depth map to measure the distance to the target from the current location. In the case of existence of multiple targets, the closest target is selected. The image plane is partitioned into five vertical non overlapping columns, labelled as left (L), forward left (FL), forward (FW), forward right (FR) and right (R) regions. Figure 7 (b) shows the resultant image after partitioning the binary image shown in Figure 7 (a) into five vertical columns. The position of the centroid of the closest target in those partitions is used to evaluate the target direction. The reference model used for the selection of the target direction is shown in Figure 8. It is selected with respect to the robot.

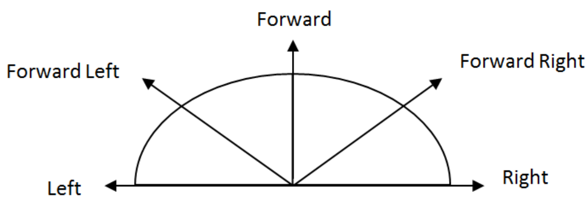

Figure 8: Reference model for target direction

TTM is implemented with five membership functions for target angle input as illustrated in Figure 9 (a). The linguistic variables used for angle input are L: Left, FL: Forward Left, F: Forward, FR: Forward Right, R: Right. The linguistic variables for the distance variable which is illustrated in Figure 9 (b) are N: Near, M: Medium, F: Far.

L angle= {L, FL, F, FR, R}

L distance= {N, M, F}

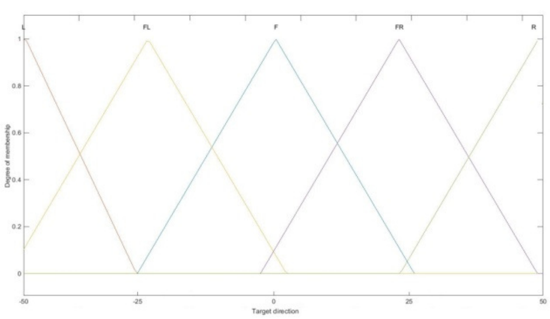

Figure 9 (a): Membership function for target direction

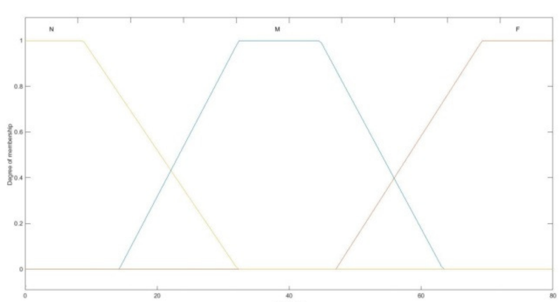

Figure 9 (b): Membership function for distance

3.4.2 Obstacle Avoidance Module (OAM)

When exploring the unknown dynamic environment as a method of mitigation the collusion with the obstacles, this method is used. The allowed turning direction is fuzzified by this process. The depth map produced by the sensor is analysed in real time. It is a 320 x 240 matrix with 11-bit values, ranging from 0 to 2047, for each pixel. The depth values produced by the sensor will be inversely proportional to the depth. Higher the depth value in a particular pixel, the 11-bit value will be lower.

A 3D depth image taken with the Xtion PRO sensor is shown below in Figure 10. For manipulation purposes the depth map is converted to 8-bit gray scale image using the following equation.

$$gray[i] = \frac{depthMap[i]*255}{(maxDepth-minDepth)} \tag{3}$$

Figure 10 (a): RGB image

Figure 10 (b) Corresponding depth image converted to gray scale

A similar portioning scheme proposed in Ravari and Taghirad (2009) is done for the gray level image constructed by depth map. After partitioning the matrix into left, forward left, right, forward right and middle regions, each partition is analysed for traversable areas by thresholding for lager depth values (i.e. smaller matrix values). This produces binary images for each partition, with 1 assigned to areas with larger depth than the threshold value and 0 otherwise. If these resultant areas to be traversed by the robot, they should be large enough to accommodate the robot. Connected component analysis, defined by Rosenfeld (1982), is used for this verification. As illustrated in Figure 11, OAM is implemented with eight membership functions. The linguistic variables are L: Left, FL: Forward Left, AL: All Left, F: Forward, FR: Forward Right, AR: All Right, R: Right, A: All area (No obstacles).

L allowed_dir = {L, FL, AL, F, FR, AR, A}

IJCSN International Journal of Computer Science and Network, Volume 4, Issue 5, October 2015
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 0.417

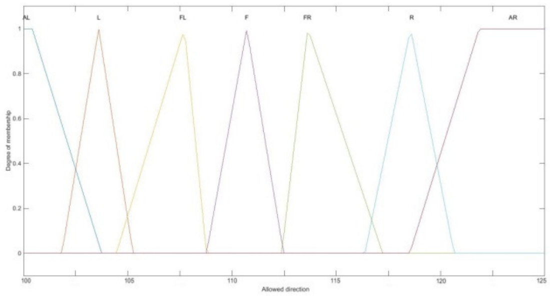

Figure 11: Membership function for allowed direction

3.4.3 Hostile Regions Avoidance Module (HRAM)

When considering industrial applications, sometimes it might be harmful for the robot to enter hostile areas such as wet areas or areas with higher temperature. Such hostile regions are represented using blue laser beams and similar to processing the red colour beams, RGB colour map is converted to HSV and then it is thresholded for blue colour. After the de-noising procedure, the binary image is inversed to get the friendly directions.

Similar to OAM module, the friendly directions are represented using eight membership functions. It's further illustrated by Figure 12. The linguistic variables are L: Left, FL: Forward Left, AL: All Left, F: Forward, FR: Forward Right, AR: All Right, R: Right, A: All area

L friendly_dir = {L, FL, AL, F, FR, AR, A}

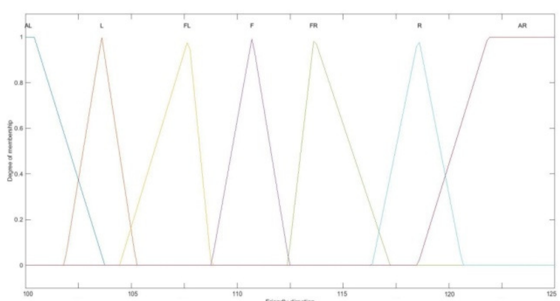

Figure 12: Membership function for friendly direction

3.4.4 Defuzzification Process

The fuzzy sets defined in the above steps are defuzzified and converted to a final exact output value representing the moving direction of the robot using Centre of Area defuzzification technique. The output variable is represented with five membership functions (Figure 13) for moving direction. Some sample fuzzy rules are illustrated in table 1.

L moving_direction = {L, FL, F, FR, R}

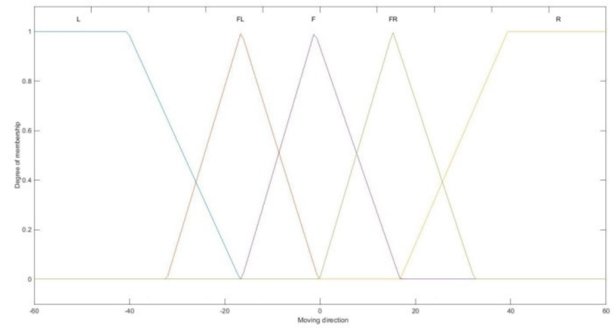

Figure 13: Membership function for moving direction

Table 1: Example fuzzy rules

#	Target Direction	Distanc e	Allowed directio n	Friendly directio n	Moving direction
1	F	N	F	F	F
2	F	M	AL	L	L
3	F	M	FL	AL	FL
4	F	M	AL	AL	FL

3.4.5 Map Generation Process

The moving directions along with the distance travelled are queued for the purpose of map generation. OpenGL (OpenGL.org 2014) libraries and glut32 libraries are used

in generation of the map. 2D Cartesian grid is loaded as the background image and the library draws the path on top of that. The output will be a scaled path which was followed by the robot (Figure 14).

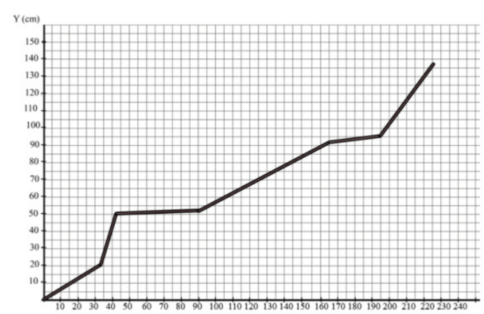

Figure 14: Generated map

4 Experimental Results

A C++ serial communication library is implemented to communicate with the Arduino microcontroller (Arduino.cc 2014) which controls the L298N dual H bridge motor diver. RP5 robot platform (Pololu.com 2014) is used with two servo motors. The output variable, moving direction will indicate what functions should be invoked among the set of functions that resides in the microcontroller. The function calls are transmitted serially between host computer and microcontroller.

Several experiments are designed to test the accuracy of each module separately and the combined operation of the robot.

4.1 Static Obstacles without A Target

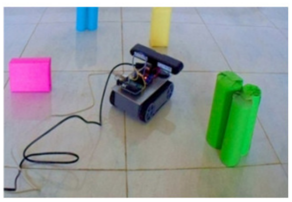

Figure 15 (a): Environment with static obstacles

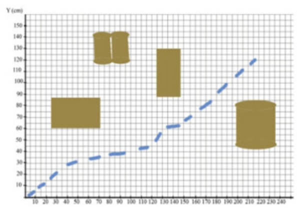

Figure 15 (b): Path taken by the robot

The mobile robot navigates through four static obstacles in the environment without a specified target (Figure 15 (a)). These obstacles have different shapes and sizes, two are rectangular and others are cylindrical shape. The produced path map is superimposed with the surrounding and the accuracy of the obstacle avoiding module is measured. The output is illustrated in Figure 15 (b).

4.2 Dynamic and Static Obstacles with A Specified Target

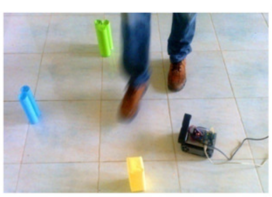

Figure 16 (a): Dynamic environment with a specified target

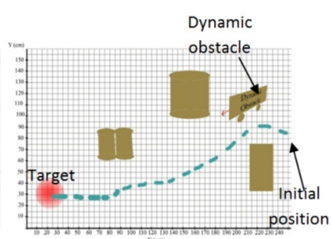

Figure 16 (b): Path taken by the robot

In this experiment one rectangular static obstacle is replaced with a dynamic obstacle and a target is specified. A researcher has been used as the dynamic obstacle. As shown on Figure 16 (a) the laser dot is placed 230cm away from the robot. In the first few iterations robots sensor doesn't pick the target. Therefore it is in the wondering mode. After reaching 110cm to the target, it recognises the target and switches to the target tracking mode. The path taken by the robot is shown in the Figure 16 (b).

4.3 Hostile Environment with a Specified Target

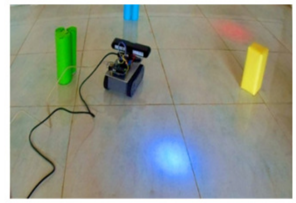

Figure 17 (a): Hostile environment with one target

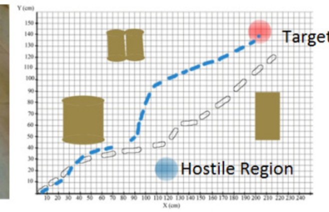

Figure 17 (b): Path taken by the robot

One hostile region is added to the previous setup in 4.1 (Figure 17 (a)). As the robot prioritises the hostile region avoidance than target tracking, the deviation of path is clearly visible. Figure 17 (b) elaborates these deviations. The path taken by the robot in the previous experiment is shown in doted lines. As the robot enters the hostile area, it prioritizes hostile region avoidance over target tracking. But later there is no significant deference in the two paths.

4.4 Hostile Environment With Two Targets

Figure 18 (a): Hostile environment with two targets

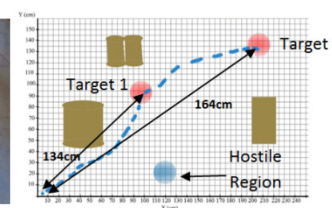

Figure 18 (b): Path taken by the robot

For the same environment described in 4.3, another target is added (Figure 18 (a)). The first target is kept 134cm from the initial position and the next in 264cm. After reaching the first target the robot moves to its next goal. Figure 18 (b) illustrates the robots path. The target tracking module selects the closest target in the existence of multiple target. Therefore though the robot sees both targets, it only selects the closest target first. After completely visiting the target 1 it moves towards the target 2. Due to the nature of the problem, it is hard to objectively evaluate the performance of the proposed methods. To say where the clear movable path exists is a matter of definition and will probably vary depending on the observer. Taking all this into account we defined a measurement called Movement Error (ME) for the goal of quantifying the accuracy of our algorithm. Movement error (ME) is the average Euclidian distance from the sample points of the optimal path to the path followed by the robot, irrespective of whether the points are above or below the optimal path.

$$\mathrm{ME} = \frac{\sum_{i=1}^{n} \sqrt{(x_{i,op} - x_{i,rp})^2 + (y_{i,op} - y_{i,rp})^2}}{\text{number of sample points (n)}} \quad (5)$$

Where $x_{i,op}$, $x_{i,rp}$ are the locations of the ith sample point in the optimal path (op) and robot path (rp) in x Cartesian axis and $y_{i,op}$, $y_{i,rp}$ in y Cartesian axis.

The optimal path is obtained by manual plotting of the path on top of a Cartesian grid by the human observer. To eliminate the human error in this process we have used 3 observers and the average path for each experiment plotted by these 3 individuals is taken as the optimal path. In order to obtain the path followed by the robot we have used the generated path map from the custom written graphics application. The movement error and time taken for robot to complete each experiment is illustrated in the following table.

Table 2 :Movement error and times per run

Experiment	Movement Error (cm)	Time per run (s)
A	3.9	12.5
B	2.7	15.1
C	7.8	13.7
D	5.0	17.2
Average	**4.8**	**14.6**

We conducted an additional experiment to compare proposed system's ability against the method proposed in Peasley (2013). According to Peasley et. al their proposed method is capable to avoid obstacles at a high rate of speed. But the approach is limited for navigations in a fixed rectangular region. We tested our approach under similar conditions, at translation velocity from 0.2 m/s to 0.8 m/s and rotation velocities from 2.8 rad/s to 5.6 rad/s. The robot successfully avoided all obstacles which were avoided as they had been at slower speeds. This verifies the applicability of the proposed method for all unstructured environments without any additional constraints. Furthermore its none dependency on prior knowledge of the environment, in contrast to methods such as Bajones (2014) and Biswas (2012), ensures the applicability of the proposed method in exploration applications such as rescue operations.

When comparing our work with the other relevant fuzzy navigation systems, our system has gained its performance due to its fuzzy logic architectural design. First when comparing our system to subsumption, which was first introduced by Brooks (1986), in subsumption all of the behaviours run concurrently and an arbitration scheme will choose a one behaviour out of all behaviours. One of the problems associated with subsumption is that the arbitration technique employed only allows a single behaviour to be active at any given time. While this is satisfactory in many situations, there are times when combination of two or more behaviours is required. As in our case navigating towards a target and avoiding obstacles and hostile regions.

Yen (1995) has enhanced the subsumption idea through a command fusion module which combines each preference from separate subsystems and defuzzify it to produce a crisp output value. They have only shown that results through a simulation procedure for robot to track a target while avoiding obstacles. In our study we have extended the idea for 3 subsystems, providing the necessity of each, and verify with actual experimental results that the idea of command fusion can be extended to any number of sub systems when considering mobile robot navigation. Additionally, we contribute to the field of mobile robotics, by demonstrating how computer vision based perception schemes can be introduced as an alternative to traditional sensing mechanisms.

5. Conclusion

This paper addresses the navigational challenges that arise in settings where mobile robots move in an unstructured environment. We have proposed a robust fuzzy logic based navigation control algorithm and a novel framework for the integration of computer vision based sensing mechanism for mobile robots. The proposed work intends to introduce robots into open, real world environments and navigate them intelligently with minimal human intervention. The study has addressed the two main issues in robot path planning. Reliable reactive obstacle and hostile region avoidance to guarantee safe operation, and smooth path planning that allows to dynamically adapt environment information with the motion of surrounding persons and objects.

The fuzzy controller was tested to perform collision-free navigation toward the given goal(s). Special care was taken to model the environment so as to be as close to real world. The experimental results have shown that the proposed architecture provides an efficient and flexible solution for light autonomous differential mobile robots, to avoid obstacles and hostile regions when navigating in structured and unstructured environments. Furthermore results suggest that low-cost 3D camera can be used to replace higher cost imaging systems in mobile robots.

With comparisons to previous studies we have verified that the proposed method has outperformed the current state-of-the-art fuzzy logic based navigational techniques for mobile robot target tracking and obstacle avoidance. The algorithm exhibits higher degree of accuracy, efficiency and flexibility when adopting to challenging environments those real world scenarios can offer. In future we will be extending the fuzzy algorithm to a neuro-fuzzy control system where higher degree of flexibility in automatic object recognition is possible. We believe this will enable us to answer more questions such as what happens when obstacles form a cluster and the target is directly behind those obstacles. At the moment when an obstacle is right in-front, the robot gets into its wandering mode. In proposed system, for each iteration all the subsystems are running concurrently. The robot will be on that mode only for that iteration of the algorithm. It's in that state because the target is covered by obstacle. In the next iteration, if it can see the target, it'll switch to target tracking mode.

References

[1] Araiza D., Tony J. (2010) Biologically Inspired Controller for the Autonomous Navigation of a Mobile Robot in an Evasion Task, *World Academy of Science, Engineering and Technology*, Vol:4, August 2010

[2] Arduino Uno micro controller, Available from: <http://arduino.cc/en/Main/arduinoBoardUno>. [April 2014]

[3] Asus Xtion Pro, Available from: <www.asus.com/Multimedia/Xtion_PRO_LIVE/> [April 2014]

[4] Bajones M., Einramhof P., Wolf D. (2014) RGB-D sensor setup for multiple tasks of home robots and experimental results, *Intelligent Robots and Systems*.

[5] Benavidez P., Jamshidi M. (2011) Mobile Robot Navigation and Target Tracking System, *6th International Conference on System of Systems Engineering, Albuquerque, New Mexico, USA,* June 2011

[6] Biswas J., Veloso M. (2012) Depth Camera Based Indoor Mobile Robot Localization and Navigation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1697-1702, May 2012.

[7] Brooks, R. A., "A Robust Layered Control System for a Mobile Robot." *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, 1986, pp. 14-23.

[8] Choppin S., Jonathan W. (2013) The potential of the Microsoft Kinect in sports analysis and biomechanics ; *Sports Technology,* Published online, 23 Aug 2013.

[9] Karthiga M., Samuel S.J. (2014) An Improved Path Planning REB Mechanism for Rescue Robot in Unmanned Area*, International journal of scientific research*, March 2014

[10] Kloss G. K., Shin S. (2009) Dynamic colour adaptation for colour object tracking, *International conference on Image and Vision Computing New Zealand*, pp. 340-345.

[11] Kolarow A., Brauckmann C. (2012) Vision-based Hyper-Real-Time Object Tracker for Robotic Applications, *International Conference on Intelligent Robots and Systems*.

[12] Lecorné S., Weitzenfeld A. (2004) Robot navigation using stereo-vision; *IEEE Latin American Robotics Symposium.* October 2004.

[13] Li T. H. S., Chang S J. (2004) Fuzzy target tracking control of autonomous robots by using infrared sensors, *IEEE on Fuzzy Systems*, vol.12, no.4, pp.491 501.

[14] Li W. (1994) Fuzzy Logic Based Robot Navigation In Uncertain Environments By Multisensory Integration, *Proceedings of IEEE International Conference on Multisensory Fusion and Integration for Intelligent Systems*.

[15] Marichal G.N., Acosta L. (2001) Obstacle avoidance for a mobile robot: A neuro-fuzzy approach, *Fuzzy Sets and Systems, Elsevier Science*.

[16] Marzouqi S., Jarvis R. (2014) Hiding strategies for a mobile robot in hostile obstacle strewn environments, *Cambridge University Press*.

[17] Mohammed F, Ramdane H. (2013) Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment, *International Journal of Advanced Robotic Systems*. January 2013.

[18] Naaz S., Alam A. (2011) Effect of different defuzzification methods in a fuzzy based load balancing application, *International Journal of Computer Science Issues*, Vol. 8, Issue 5, No 1, September 2011

[19] OpenCV library computer software, Available from: <www.opencv.org> [April 2014]

[20] OpenGL library, computer software, Available from: <www.opengl.org/resources/libraries/glut/> [April 2014]

[21] OpenGL library, computer software, Available from: <www.opengl.org> [April 2014]

[22] OpenNI library, computer software, Available from: <www.openni.org> [April 2014].

[23] Pai N.S, Hsieh H.H. (2012) Implementation of Obstacle-Avoidance Control for an Autonomous Omni-Directional Mobile Robot Based on Extension Theory, *Sensors* 2012

[24] Payton D.W., Rosenblatt J.K. (1990) Plan Guided Reaction*, IEEE Trans. Systems, Man and Cybernetics,* Nov 1990.

[25] Peasley B., Birchfield S. (2013) Real-Time Obstacle Detection and Avoidance in the Presence of Specular Surfaces Using an Active 3D Sensor, *IEEE Workshop on Robot Vision (WORV).*

[26] Rashid R., Elamvazuthi I. (2010) Differential Drive Wheeled Mobile Robot Control Using Fuzzy Logic Techniques, *Proceedings of the 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, pp. 51–55.

[27] Ravari A.R.N., Taghirad H.D. (2009) Vision-based fuzzy navigation of mobile robots in grassland environments, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1441-1446.

[28] Ronald C., Robin R. (1990) Autonomous navigation in manufacturing environment, *IEEE Transactions in robotics and automation,* August 1990.

[29] Rosenfeld A., Kak A. C. (1982) *Digital Picture Processing*, vol. 1. Academic Press, Orlando, USA.

[30] RP 5 Robot platform, Available from: <http://www.pololu.com/product/1060> [April 2014]

[31] Rusu C. G., Birou I. T. (2010) Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors, *10th International Conference on development and application systems, Suceava, Romania*, May 2010.

[32] Saffiotti A. (1997) Fuzzy Logic in Autonomous Robotics: behaviour coordination, *IEEE conference in fuzzy systems*. July 1997.

[33] Santos D., Correa O., Fernando D. (2012) Mobile Robots Navigation in Indoor Environments Using Kinect Sensor, *Second Brazilian Conference on Critical Embedded Systems*.

[34] Thrun S., Montemerlo M. (2009) The Robot that Won the DARPA Grand Challenge, *Journal of Field Robotics*, vol. 23, n. 9, pp. 661-692, June 2006.

[35] Xi Li , Byung J. (2013) Design of Obstacle Avoidance System for Mobile Robot using Fuzzy Logic Systems, *International Journal of Smart Home,* Vol. 7, no. 3, May 2013.

[36] Yen, J., Pfluger, N., "A Fuzzy Logic Based Extension to Payton and Rosenblatt's Command Fusion Method for Mobile Robot Navigation." *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 6, pp. 971-978, June 1995

First Author

Tharindu Fernando is currently a final year student following a BSc computer science special degree program in Faculty of science, University of Peradeniya Sir Lanka. His research interests include, Artificial intelligence, Machine Learning, Computer vison and human computer interaction.

Second Author

Harshala Gammulle is currently an undergraduate student following a BSc computer science special degree program in Faculty of science, University of Peradeniya Sir Lanka. Her research interest include, Digital forensics, Artificial intelligence and image processing.

Third Author

Chamila Walgampaya earned his B.Sc. in Computer Engineering with honours in November 2001 from the Faculty of Engineering, University of Peradeniya, Sri Lanka. He earned his Ph.D. in August 2011 from the School of Engineering at the University of Louisville. For his Ph.D., he worked on developing a novel click fraud detection and prevention system for Internet advertising. His disseration titled is "Click Fruad: how to spot it? how to stop it?" He is currently a lecturer in the Department of Engineering Mathematics, University of Peradeniya.