Mobile robot navigation and target tracking system

Article · June 2011

DOI: 10.1109/SYSOSE.2011.5966614

2 authors:

Patrick Benavidez
University of Texas at San Antonio

Mo M. Jamshidi
University of Texas at San Antonio

Proc. of the 2011 6th International Conference on System of Systems Engineering, Albuquerque, New Mexico, USA - June 27-30, 2011

Mobile Robot Navigation and Target Tracking System

Patrick Benavidez
Department of Electrical and Computer Engineering
The University of Texas at San Antonio
San Antonio, USA
gax646@my.utsa.edu

Mo Jamshidi, Ph.D., Lutcher Brown Endowed Chair
Department of Electrical and Computer Engineering
The University of Texas at San Antonio
San Antonio, USA
mo.jamshidi@utsa.edu.

Abstract—This paper presents the framework for the navigation and target tracking system for a mobile robot. Navigation and target tracking are to be performed using a Microsoft Xbox Kinect sensor which provides RGB color and 3D depth imaging data to an x86 based computer onboard the robot running Ubuntu Linux. A fuzzy logic controller to be implemented on the computer is considered for control of the robot in obstacle avoidance and target following. Data collected by the computer is to be sent to a server for processing with learning-based systems utilizing neural networks for pattern recognition, object tracking, long-term path planning and process improvement. An eventual goal of this work is to create a multi-agent robot system that is able to work autonomously in an outdoor environment.

Keywords-mobile robot; fuzzy logic; path planning; neural networks; learning system

I. INTRODUCTION

Many control schemes have been presented for navigation and target tracking for use on mobile robots. Control schemes used for navigation include conventional controller designs such as PD and PID controllers, and also controllers taken from intelligent control systems with fuzzy logic [1] and neural networks [2], [3] being the most commonly used. In the case of navigation, most algorithms plan a route ahead of time based on what data is available on the robot's working environment and require minor real-time corrections based on local disturbances not identified in the original data on the environment. Therefore navigational algorithms can incorporate lengthy simulations or other non-real time operations running in tandem with the basic operations of the robot. Parallel processing of navigation and other critical operations allow a robot to perform obstacle avoidance and target tracking in real time, leaving lengthy computations to other agents in a system. Separating these operations is common in many mobile robot systems that allow distributed processing where the robot does not necessarily need to calculate its own path, such as sports playing robots as in [4] and [5]. Additionally, separation of navigation and other critical operations is performed using multiple computers or computing servers located either remotely or fixed on the robot.

This work was funded by scholarship funds made available by The Valero Energy Corporation and support from a Lutcher Brown Endowed Chair

Several well known examples of unmanned ground vehicles were seen in the DARPA Grand Challenges where teams of researchers entered robotic automobiles in an autonomous race to successfully navigate a particular type of area or terrain. Vehicles in the Grand Challenge utilized either vision or depth information, or a fusion of both sensor data to navigate unstructured environments for which the researchers had limited previous knowledge. Examples of teams from the Grand Challenge using both laser range finders and cameras are [6] and [7], teams using only laser range finders included [8], and teams using only cameras included [9].

Besides the DARPA Grand Challenge, many researchers have implemented similar systems for autonomous ground vehicles including [10]-[13]. Common to some of the aforementioned examples of systems using camera information to navigate an environment is the use of the Hough transform or similar methods to identify straight line features in an image. The straight line information is used in [3] and [12] to identify landmarks in images to compare between frames in order to track the progression of the robot through the environment. Also common to most works is the identification of the traversable region in an RGB or depth image. Various methods have been used and proposed based on the characteristics of the environment being traversed and usually rely on segmentation of the image into distinct areas for sub-processing [1], [10], [12], [13], [14]. The following sections investigate navigation, ground plane identification and fuzzy logic control of a robot for target tracking. The sections are ordered as follows: navigation of a mobile robot, object recognition, supervisory via remote computing server, and navigation and target tracking.

II. NAVIGATION OF A MOBILE ROBOT

Navigation of a mobile robot consists of path planning and heading changes towards the target destination. Navigation of a mobile robot can be assisted by systems covered in computational intelligence theory. Path planning and obstacle avoidance are considered here in this section.

A. Dynamic Model of Differential Drive Robot

The basic dynamic model of a robot is required for its control. In this case we describe a model of a differential drive robot as shown in Fig. 1. The terms $v_\ell(t)$, $v_r(t)$, $v(t)$, $\omega(t)$ and L in (1) below refer to the velocity of the right set of tires, velocity of the left set of tires, translational velocity of the robot, rotational velocity of the robot, and the length between the left and right sets of tires.

$$v(t) = (v_r(t) + v_\ell(t))/2 \tag{1}$$

$$\omega(t) = (v_r(t) - v_\ell(t))/L \tag{2}$$

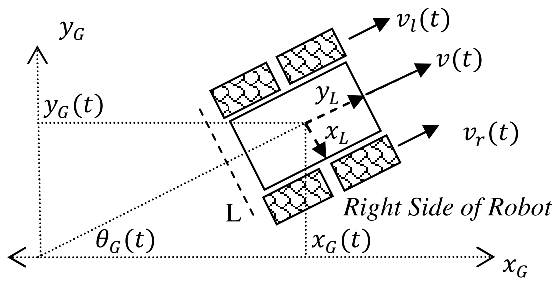


Figure 1: Model of Differential Drive Robot

This dynamic model is used later in Section IV for control of the robot.

B. Object Recognition

The purpose of object recognition in this paper is two-fold, firstly to identify objects to track in the environment and secondly to identify free space in the space in front of the robot for navigation purposes. The following sections detail the sensor capable of generating RGB-D images, and how we plan to use implement image processing and 3D depth image processing.

1) RGB-D Sensor: The cost and power requirements of 3D depth imaging sensors has long been a prohibitive factor in its deployment on small cheap mobile robots. Now with Microsoft's Xbox Kinect, the cost of 3D depth imaging has gone down by at least a factor of 10 (comparing to laser range finders costing greater than 1500 USD) and the power requirement has been reduced due to the lack of moving parts, with exception of the motor for tilt control, such as those in 3D laser scanners commonly used on mobile robots. The Kinect is an RGB camera along with a 3D depth ranging sensor that works through infrared light [16]. This sensor provides access to two images, one RGB image and also a 3D depth image. Use of a combination of both of these images provides a fair level of capabilities for object recognition. Software used to interface with the Kinect to obtain access to its sensors and motor is provided by an open source software project called OpenKinect [18]. OpenKinect was used in this paper to acquire the depth and RGB images presented later in this section.

2) Color Image Recognition: In this section use of the rg-chromaticity color space, adaptive color processing edge detection and color segmentation are proposed as being part of the color image recognition routine. This routine will be used for target tracking and navigation for the mobile robot.

a) rg-Chromaticy Color Space: One of the main problems in color image processing is the variablility of colors due to local intensity shifts in the environment. To abate this problem, a color space called rg-chromaticity is used by [4], [5] to remove the light intensity from the determination of the color. Transforms to the rg-chromaticity space are listed below in (2)-(5).

$$r_{cr} = R/(R + G + B) \tag{3}$$

$$g_{cr} = G/(R + G + B) \tag{4}$$

$$i_{cr} = (R + G + B)/(3 * 255) \tag{5}$$

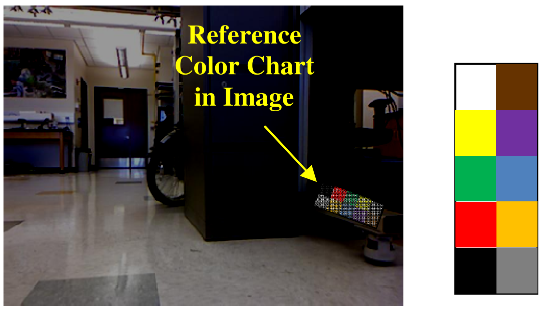
With rg-chromaticity, primary and secondary colors can be isolated into specific boundaries in the red and green chromaticity spaces with a fair amount of reliability. Given these boundaries, fuzzy logic can be applied to identify certain colors in systems of robots which rely on color identification as seen in [4] where a color camera is used as a global vision source to identify paths that the robots take and also location of their target (a golf ball). In the system, both the robots and the targets are marked with a specific color to aid in the identification of the objects. Example empirical values for rg-chromaticity determined by researchers in [18] are listed in Table 1.

TABLE I. SELECTED EMPIRCAL RESULTS FOR RG-CHROMATICITY SPACE COLOR IDENTIFICATION [18]

COLOR	$r_{cr,min}$	$r_{c,max}$	$g_{cr,min}$	$g_{cr,max}$
Red	0.6	0.7	0.088	0.176
Orange	0.523	0.619	0.238	0.285
Blue	0.1	0.2	0.3529412	0.441
Black	0.2	0.3	0.2647059	0.352

Examining results of [18] in Table 1, identification of colors in the rg-chromativity space is appropriate for fuzzy logic, as suggested in [18], as each color is defined within a certain region which may overlap.

b) Adaptive Color Processing: In a system which is not used in a well defined area with a constant light source, an adaptive color processing algorithm is better suited for use than a non-adaptive one. In [4] researchers experimented with shifts in light intensity. In their experiments, they placed a known color reference chart in the image similar to the one depicted below in Fig. 2 (b). With the color reference chart present in the view of a camera, the researchers varied the intensity of light in the room while capturing image frames. In processing of the images, they used a dynamic correction algorithm which accurately corrected low lit colors to their original color based on the color reference pattern present in the image. With the original colors retrieved, they reliably tracked a robot through the images. The neural networks would be trained on the reference color pattern and would be used to identify the reference chart in the image frame. From the identified patterns, the transformation between the pattern in the image and the color reference chart would be known based on the difference between the colors and intensities.



(a) Image Frame (b) Color Reference Chart

Figure 2: Image Frame Containing Reference Color Chart and Original Color Reference Chart

3) Color Segmentation: The combination of using the rg-chromaticity color space and light intensity correction provide two components necessary for color image recognition. An image can be further segmented into components containing colors similar to primary, secondary, and tertiary colors of varying intensities. With the color segmented image, candidates of objects known to be a certain color can be isolated in an image frame and examined further.

C. 3D Depth Image Recognition

A 3D depth image taken with the Kinect sensor is shown below in Fig. 3. In the image, it is apparent that the image provides a higher level of distance definition compared to that of a color RGB image taken by the RGB camera as seen above in Fig. 2. The traversable area can be identified in the depth image using a 2D gradient and a 2D log filter, as shown in Fig. 4.

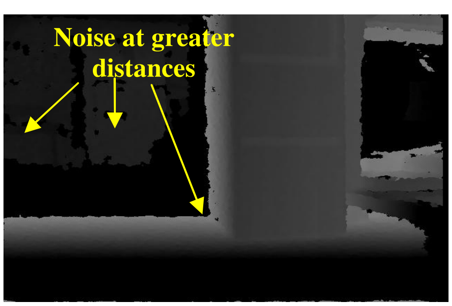


Figure 3: 3D Depth Image; Dark Areas Indicate Close Distances in Foreground and Areas of Noisy Data in Image Background

Planned processing of the depth image involves finding the traversable area (shown in Fig. 4) and the planning a route using the depth image. One method used in [1], in which an RGB image was segmented into three vertical columns – center, left and right, will be used to identify regions in the depth image for obstacle avoidance.

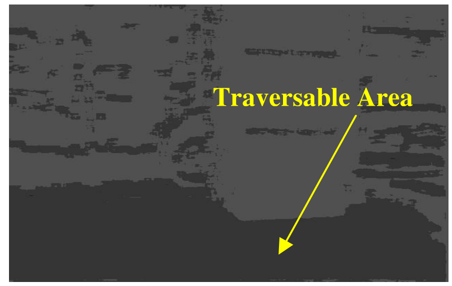


Figure 4: Log-Filtered Gradient of 3D Depth Image; Solid Dark Gray Patch Indicates Flat Traversable Region of Image (Indicated in Yellow)

III. AUTONOMOUS SUPERVISORY OF ROBOT VIA REMOTE COMPUTING SERVER

During the course of an experiment with an autonomous robot, it is crucial to record data on the robot's interaction with the environment in order to improve its capability to be fully autonomous. Real-time learning by the robot is unlikely given the computing power a typical robot has and the time duration for learning algorithms to converge. Instead, we propose the use of a server to collect data from a robot in the field, to aggregate the data to find undesired trends in behavior to certain stimuli, and to provide a supervisory role in the control of a robot.

A. Data Collection

This section assumes that a robot has the ability to communicate to the computing server either over a Wi-Fi connection to a wired network or a 3G/4G modem with connection to the server over the internet. At regular intervals data collected by various processes running on the robot's computer is uploaded to the server over a network. Data uploaded to the server is to be then stored and processed. Depending on the type of data sent to the server, a compressed container should be used to send the data when it is split between multiple files. Distance between backup intervals is dictated by the data rate and latency properties of the network in addition to the time sensitivity of the data collection. In the case that the server provides time sensitive feedback that the robot needs in order to make an action, the backups will be spaced close together. Otherwise data transactions are spaced further apart.

B. Data Aggregation

Aggregation is to be performed on data uploaded to the server to identify many different trends. Trends discovered in reactions to various stimuli can be used to identify when control algorithms are working properly and can be used to improve them when they are not. Logs including documented reactions to sensor data at timestamps matching the data can be examined by a dedicated process running on a server.

C. *Supervisory Role in Control of Robot*

A supervisory role in the control of a robot can be performed by the remote server. Using concepts developed later in Section IV, the computing server component can recognize patterns using tools such as associative memories created with neural networks or radial basis function networks. The following is true of neural networks designed to be associative memories [19]:

- A neural network designed to be an associative memory requires one or more sets of hidden neuron layers which are trained to identify patterns in input vectors.
- Training vectors can be of any size as long as they provide sufficient information in the vector for recognition. In the case of grayscale, color or depth images, smaller compressed images are used as training vectors for the neural network as it reduces the overall time to train and identify patterns.
- After training, the network is fed newly captured input vectors (images) to be identified. The network chooses the pattern most similar to that of the newly captured image.

Design of the associative memory that provides the object tracking service to the robot is to be provided in a future research.

IV. Fuzzy Logic Control of Robot in Target Tracking Mode

A. *Generation of Fuzzy Rules*

Rules for target tracking can be generated based off of the data provided by the object tracking and recognition engine. Data that can be used for fuzzy variables include the translational and rotational velocity of the robot, and the identified target's centroid and relative size. The process of fuzzifying the input data is depicted below in Fig. 5.

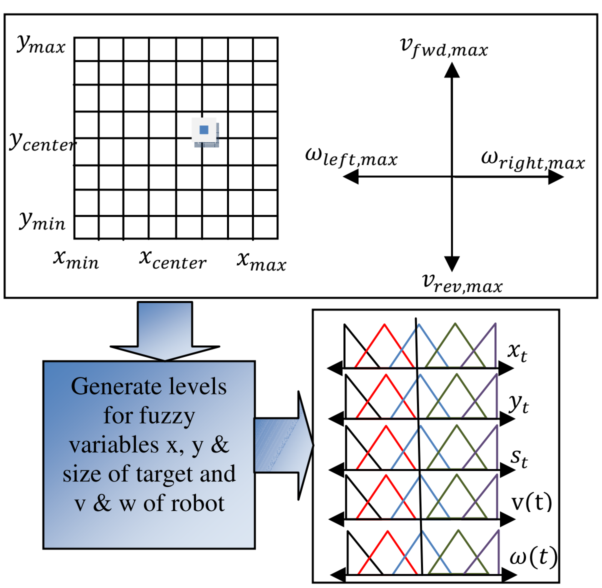


Figure 5: Generation of Fuzzy Variables for Controller

The number of levels for each fuzzy variable are chosen based on the following criteria: level of precision in movements, size of targets, range of velocities of robot.

B. *Fuzzy Controller:*

An example of a fuzzy controller is shown in Fig. 6 for the data described above in Fig. 5. The process that the fuzzy controller undergoes is as follows:

- If a target has been found and the robot is to track a target, the robot begins to follow the fuzzy controller output.
- The fuzzifier receives velocity input from the microcontroller and target data from the object recognition and tracking engine.
- The fuzzy inference engine output is fed to the microcontroller to control the motors to reach the desired translational and rotational velocities.

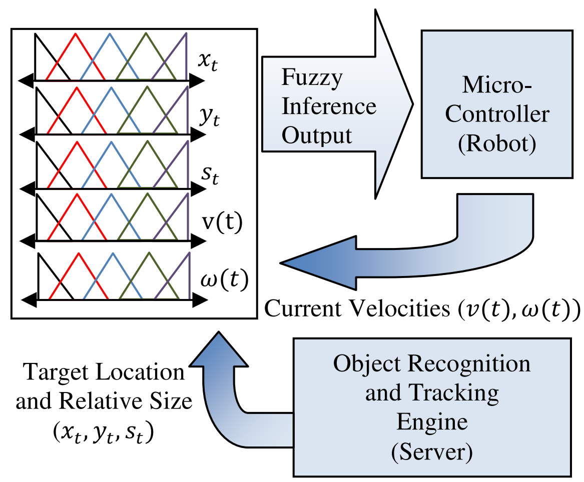


Figure 6: Fuzzy Logic Control of Robot in Target Tracking Mode

Also depicted in the figure is the existence of five levels for each fuzzified variable. Fuzzy rules generated for the system in Fig. 6 can be described as follows in Table 2 assuming the fuzzy sets include the following:

- LN: Large Negative
- SN: Small Negative
- Z: Zero
- SP: Small Positive
- LP: Large Positive

TABLE II. Example Fuzzy Rules for Controller

Rule	x_t	y_t	s_t	$v(t)$	$\omega(t)$	$v^*(t)$	$\omega^*(t)$
1	LP	LP	LN	Z	Z	LP	SP
2	LP	LP	LP	Z	Z	SN	SP
3	LN	Z	Z	Z	Z	Z	LP
4	SP	Z	Z	Z	Z	Z	SN

In Table 2, values for the location of the target are referenced to the center of the image, the relative size is referenced to the original size measurement with smaller than reference being negative, negative translation velocity being that in reverse, negative rotational velocity being rotation to the left, and $v^*(t)$ and $\omega^*(t)$ refer to the output of the controller in the next state. In the final controller, the number of fuzzy rules will likely be greater than that shown in Table II to achieve a finer level of control. Rule 3 for example describes the situation where:

- The centroid of the target being tracked is to the far left of the robot
- The robot will rotate in the next step to the left at a fast pace

If the robot rotates too quickly for the fuzzy controller to recognize the motion, this is an indication that the target has been centered and that the target is now more to the right of the robot. Another rule such as that of rule 4 may fire to correct the overshoot, which is described by the following:

- The centroid of the target being tracked is slightly to the right of robot
- The robot will rotate in the next step to the left at a slow pace

V. Conclusions

In examination of resources available in vision based robotic systems, controllers for mobile robot navigation, path planning, and target tracking are possible and have already been implemented for various robots and robot systems. Many systems have been created to allow small robots to play tournament style sports, and for robots to autonomously navigate indoors and outdoors. Problems with the more expensive systems lie in their scalability due to their use of high-end equipment and need for computing power. In this paper the combination of low-cost 3D depth and color imaging is proposed to replace higher cost imaging systems. To supply the necessary computational power we suggested use of a remote server to perform target recognition and tracking in addition to aggregating data on performance of the robot. Fuzzy logic will supply the control mechanism necessary to follow and navigate towards targets. Target selection and registration is a field not yet explored by this author and is subject of future research.

References

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

[2] S. Tangruamsub, M. Tsuboyama, A. Kawewong, and O. Hasegawa, "Mobile robot vision-based navigation using self-organizing and incremental neural networks," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2009, pp. 3094-3101.

[3] M. Meng and A.C. Kak, "NEURO-NAV: a neural network based architecture for vision-guided mobile robot navigation using non-metrical models of the environment," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* , 1993, pp. 750-757.

[4] Guy K. Kloss, Heesang Shin, and Napoleon H. Reyes, "Dynamic colour adaptation for colour object tracking," in *Image and Vision Computing New Zealand*, Wellington, 2009, pp. 340-345.

[5] K.G. B. Leong, S. W. Licarte, G. M. S. Oblepias, E. M. J. Palomado, and E.P.Dadios N. G. Jabson, "The Autonomous Golf Playing Micro Robot: With Global Vision And Fuzzy Logic Controller," *International Journal on Smart Sensing and Intelligent Systems*, vol. 1, no. 4, pp. 824-841, December 2008.

[6] R. Behringer et al., "RASCAL - an autonomous ground vehicle for desert driving in the DARPA Grand Challenge 2005," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE* , 2005, pp. 644-649.

[7] K.A. Redmill, J.I. Martin, and O. Ozguner, "Sensing and Sensor Fusion for the 2005 Desert Buckeyes DARPA Grand Challenge Offroad Autonomous Vehicle," in *Intelligent Vehicles Symposium, 2006 IEEE* , 2006, pp. 528-533.

[8] A. Bacha et al., "The DARPA Grand Challenge: overview of the Virginia Tech vehicle and experience," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, 2004, pp. 481- 486.

[9] A. Broggi, C. Caraffi, P.P. Porta, and P. Zani, "The Single Frame Stereo Vision System for Reliable Obstacle Detection Used during the 2005 DARPA Grand Challenge on TerraMax," in *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE* , 2006, pp. 745-752.

[10] S. Vitabile, G. Pilato, F. Pullara, and F. Sorbello, "A navigation system for vision-guided mobile robots," in *Image Analysis and Processing, 1999. Proceedings. International Conference on* , 1999, pp. 566-571.

[11] A. Gopalakrishnan, S. Greene, and A. Sekmen, "Vision-based mobile robot learning and navigation," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, 2005, pp. 48-53.

[12] P. Borges, R. Zlot, M. Bosse, S. Nuske, and A. Tews, "Vision-based localization using an edge map extracted from 3D laser range data," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on* , 2010, pp. 4902-4909.

[13] A. Miranda Neto and L. Rittner, "A simple and efficient Road Detection Algorithm for Real Time Autonomous Navigation based on Monocular Vision," in *Robotics Symposium, 2006. LARS '06. IEEE 3rd Latin American* , 2006, pp. 92-99.

[14] T. Low and A. Manzanera, "Ground-plane classification for robot navigation: Combining multiple cues toward a visual-based learning system," in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on* , 2010, pp. 994-999.

[15] G.N. Desouza and A.C. Kak, "Vision for mobile robot navigation: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol. 24, no. 2, pp. 237-267, Feb 2002.

[16] Stephanie Crawford. (2011, March) HowStuffWorks - Microsoft Kinect. [Online]. http://electronics.howstuffworks.com/microsoft-kinect2.htm

[17] Joshua Blake. (2011, April) Main Page - OpenKinect. [Online]. http://openkinect.org/wiki/Main_Page

[18] N.H. Reyes and E.P. Dadios, "A fuzzy approach in color object detection," in *IEEE International Conference on Industrial Technology*, Bangkok, 2002, pp. 232-237.

[19] Laxmidhar Behera and Indrani Kar, "Multi-layered Neural Networks," in *Intelligent Systems and Control Principles and Applications*. United States of America: Oxford University Press, 2010, ch. 2, pp. 41-85.

[20] Wen Shang, Xudong Ma, and Xianzhong Dai, "3D objects detection with Bayesian networks for vision-guided mobile robot navigation," in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, 2004, pp. 1134-1139.