Autonomous Computer Vision-Assisted Robotic Vehicle

Cher Hau Seng [1#], Prashan Premaratne [2]
[1]School of Electrical, Computer & Telecommunications Engineering, University of Wollongong
NSW, Australia., chs50@uow.edu.au
[2]School of Electrical, Computer & Telecommunications Engineering, University of Wollongong
NSW, Australia., prashan@uow.edu.au

Abstract

Autonomous robotic vehicle are robots which can perform certain desired tasks in a dynamic environment without human guidance to a certain degree. Most autonomous robots rely heavily on readily available off-the-shelf sensors to perform navigation. The research published here shows a glimpse of an innovative application of image processing in future robotic vehicles, which are capable of making their own decisions to avoid obstacles and plan its path in order to be truly autonomous in navigating dynamic environments. Our low cost Autonomous Robotic Vehicle (AutoRoVe) is competent of avoiding obstacles and mapping its path and move ahead using laser-beam assisted vision control system that is well capable of navigating uncharted territories. The system comprises of a an onboard 1GHz Via C3 processor with a mini-ITX motherboard and 512MB of RAM, a USB colour camera and laser tracking system that demonstrates the flavour of self navigating capability that future mobile robots may have.

1. INTRODUCTION

The primary objective of designing autonomous robotic vehicles is to perform navigation and certain tasks in unexplored environments. A high degree of autonomy is particularly desirable in areas where communication delays and interruptions are critical. For instance, space exploration rovers will require a higher degree of autonomy compared to household robots. The major area of robotic autonomy is to enable a mobile robot to manage unknown dynamic surroundings, as certain robots are strictly confined to their direct environment.

Our AutoRoVe project tries to address the current drawback of mobile robots' control systems that need frequent communications that are prone to delays and interruptions. This project designed and built an autonomous control system that has the capability to achieve intelligent decision making for navigation. The control system is implemented on a small robotic vehicle to accomplish self navigation capability.

2. AUTONOMOUS CONTROL SYSTEMS

Autonomous control systems are deployed largely in every aspect of the industry. They are commonly used in military, manufacturing and design applications in research and development laboratories or manufacturing plants. In the context of mobile robotics, autonomous control systems integrated into robots allowed them to act on their own as they have the ability to respond in a certain way to certain stimuli, independent of the controller [1]. The combination of several components such as sensors, instruments, mapping abilities and machine vision engaged into mobile robots provide artificial intelligence (AI) that could increase the productivity. For instance, a vacuum robot could help reduce the burden of human beings by cleaning up a warehouse autonomously without any supervision [2]. Mobile robots, such as the Mars Exploration Rovers (MER) currently used by NASA, could semi-autonomously explore uninhabitable venues, besides gathering scientific data on the Martian landscapes.

A. Artificial intelligence

AI is possibly the most exciting field in robotics [1] as it usually deals with uncertainty, reliability and real-time response when sensing, acting and planning are integrated into a single system [2]. An AI mobile robot has the ability to gather facts about a scenario or situation through built-in sensors or human input. The information is then compared with the stored data and the robot decides what the information signifies. Various possible actions are determined and the computer predicts the most successful action based on the collected data. Varieties of architectures have been proposed for autonomous robots, relying largely on AI techniques for merging high-level and reactive control of the mobile robot [3].

B. Mapping and navigation

Mobile robots have the option of navigating around without the aid of mapping by relying mainly on sensors. However, more sophisticated landscapes require mobile robots to build maps for navigation purposes as mapping provides solutions to obstacle avoidance by estimating the rover position and orientation and executing global path planning [2]. There are three types of commonly used maps in the mobile robot's perspective. This includes Euclidean maps that are based on absolute reference frame and numerical measures of position and dimensions of objects; quad trees, which are recursive data structures for parsimonious representation of occupancy grids [4]; and topological maps that contain objects and connectivity information, using neither metric nor geometric information [2], [4].

Global Positioning Systems (GPS) were also introduced to autonomous control systems during the development of the MERs [5], which include a construction of the Mars global terrain database. A rover equipped with a GPS receiver would not need to undertake a lost-search algorithm since the rover location could always be determined from the positions of three GPS satellites based on the triangulation algorithm [5].

C. Sensors

Sensors used in autonomous control systems could be classified into two classifications, Classification 1 and Classification 2 [6]. In Classification 1, sensors are categorized based on the output or feedback that is provided by the internal or external sensors. Internal sensors are used to provide feedback about the internal state of the mobile robot, such as battery levels and wheel positions. External sensors on the other hand provide feedback on the external state, such as the surrounding temperature and light intensity [6].

Classification 2 is used to categorize sensors actively and passively by their ability to emit and receive energy from their environments. Passive sensors used in mobile robots have the ability to receive energy from the environment whereas active sensors make observations by emitting energy [6].

The AutoRoVe project utilizes a laser range finder as an active laser emitter and a sensor. The combination of a laser emitter, simulated by a laser pointer, and a sensor, simulated by a USB web-camera, provides accurate distance measurements and it could be configured for multi-angular measurements through a rotating mirror [6]. Besides that, the laser beams could be easily distinguished from ambient light, although there could be atmospheric influences on laser lights [6]. Similarly, laser emitters could also be used with cameras to avoid obstacles and to navigate around, as implemented in the Rocky IV Mars rover prototype by NASA back in 1995 [7].

3. OVERVIEW OF SOFTWARE DESIGN

Initially the project focused on developing the control systems based on MATLAB controlling the hardware so that the AutoRoVe can traverse a pre-assigned path. For an autonomous control system to have the ability to navigate by itself, text based commands are more suitable to be used than graphical interfaces. However, in order to test the control system effectively and conveniently, Visual Basic Graphical User Interface (GUI) was written to control the parallel port bypassing Matlab. This step had the added advantage of being able to control the system with extremely moderate processing requirements (i.e. Pentium 100 MHz has enough capability). The overall software design architecture is shown in Fig. 1 below. As shown in Fig. 2 below, the software was designed to allow a user to interact with the system using a GUI.

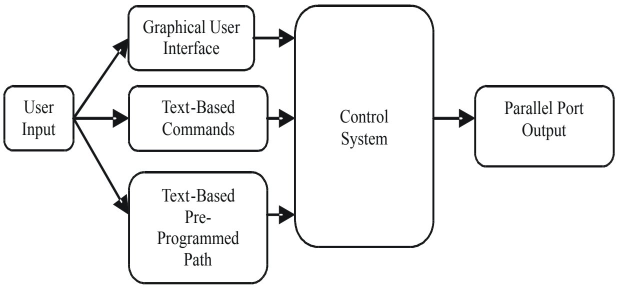

Fig. 1: Overall Software Design Architecture

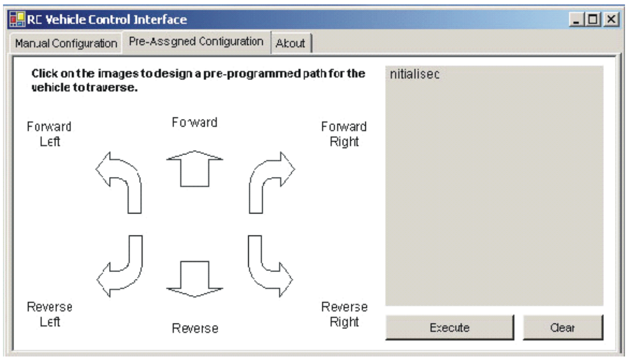

Fig. 2: GUI Control Pad

Using the GUI, the users have the flexibility of setting the range of duration and resolution traversal for the mobile robot. A user could also assign a traversal path of any length (duration), and by the executing the command, moves the vehicle in the assigned path.

The software was then fine-tuned to allow the mobile robot to navigate by itself without any interaction with the user. The user has now the flexibility to semi-autonomously control the mobile robot or to set it to autonomous mode, by executing a command line in MATLAB.

4. HARDWARE DESIGN

AutoRoVe comprises of plastic tracks driven by two DC brushless motors. The on-board computer sends the appropriate commands through the parallel port's eight (8) data pins. Out of these eight (8) independent signals, six (6) data pins are used for forward, backward, left and right movements. Two more data pins are used for revolving the camera and the laser beam mounted platform. These data pins are then connected to an npn transistor-based switching circuit that controls six (6) relays to control the DC motors in the vehicle.

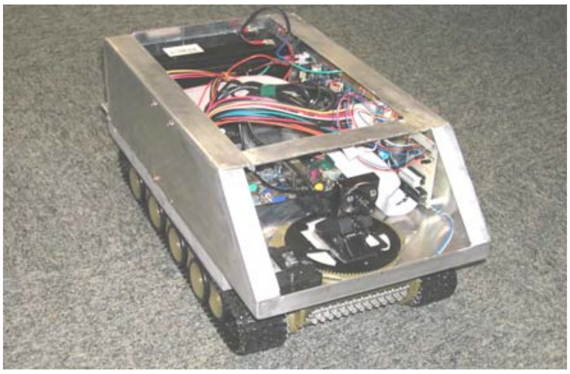

Fig. 3: Image of the AutoRove without the Top Cover

The most significant hardware component in this project is the development of the laser-beam assisted obstacle avoidance system. As shown in Fig. 4, a simple triangulation technique is used to calculate the distance from the camera and the laser beam to the obstacle. The obstacle distance *d* is calculated as

$$d = \frac{h}{\tan\theta} \quad (1)$$

and

$$\theta = pfc * rpc + ro \quad (2)$$

where *pfc* is the *Number of pixels from Centre of Focal Plane*, *rpc* is the *Radius per Pixel Pitch* and *ro* is the *Radius Offset*, which compensates for the alignment errors.

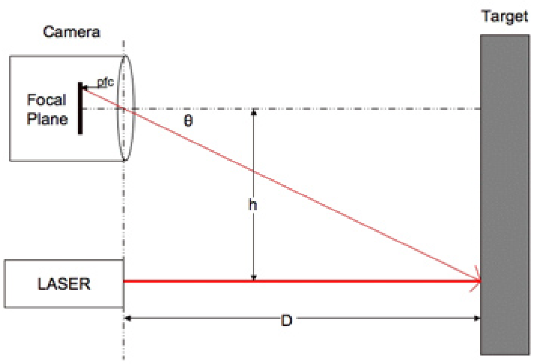

Fig. 4: Laser-beam Triangulation Diagram

A lot of sample points were obtained before deriving the linear relationship between angle (θ) and the number of pixels from centre of focal plane (pfc). The laser range finder technique was fine-tuned and the following test results have been obtained:

TABLE 1: ANGLE MEASUREMENTS USING TRIANGULATION

Pixels from Center	Theta	
	Actual	Calculated
124	0.294003	0.300704
92	0.266040	0.245738
65	0.214642	0.199361
61	0.179707	0.192491
46	0.154474	0.166726
36	0.135416	0.149549
19	0.120524	0.120349
16	0.108571	0.115196
6	0.098768	0.098019
-3	0.090585	0.082560
-7	0.083650	0.075689
-10	0.077700	0.081899
-14	0.072539	0.071291
-15	0.068020	0.068639
-19	0.064030	0.058031
-33	0.018472	0.020903

TABLE 2: DISTANCE MEASUREMENTS USING TRIANGULATION

Distance		% Error
Actual	Calculated	
18.000	17.575	-2.36%
20.000	21.730	8.65%
25.000	26.974	7.90%
30.000	27.962	-6.79%
35.000	32.385	-7.47%
40.000	36.171	-9.57%
45.000	45.066	0.15%
50.000	47.101	-5.80%
55.000	55.423	0.77%
60.000	65.863	9.77%
65.000	71.867	10.57%
70.000	66.397	-5.15%
75.000	76.318	1.76%
80.000	79.276	-0.90%
85.000	93.810	10.36%
295.000	260.696	-11.63%

From these experiments, we have determined the distance calculation with an accuracy of 5cm is possible, which is adequate for decision making by the control system.

A. *Computer-based decision making*

Machine vision or computer vision extracts high-level information from visual scenes. Image processing is then used to process that input to produce a new visual representation by using certain methods and algorithms [7]. Machine vision could be further categorized into three levels. These levels include low-level vision, intermediate vision and high-level vision [7]. Low-level vision could determine local image properties via smoothing to establish edges, colour and textures of the captured images. Intermediate vision has object recognition capabilities to determine generic scene attributes. This could be used for model matching, boundaries, regions and surface detection. Machine vision has also been integrated into mobile robots since the 1960s [7]. High-level visions are commonly integrated onto upcoming Mars rover prototypes at the NASA Jet Propulsion Laboratory as the advancement in vectored processors improves the performance of stereo vision implementation [8]. Stereo vision was desirable for the MERs since it provides long-range high resolution images with its passive sensors [2], besides allowing the rover to percept depth, locate and classify objects [1]. This enables the rover to navigate around avoiding obstacles, making it more ideal for dynamic environments.

In this project, we have used low level vision that determines the edges, corners and major obstacles in the path. The images will be captured into MATLAB using a video camera and then converted to a binary image using appropriate threshold. Edge detection is used on this image and major objects in the scene is identified and tagged with their distances to the camera. The system will determine the path to be traversed using the unobstructed paths that offer the largest opening between obstacles. This decision making process takes place every ten (10) seconds and few centimeters will be traveled within this period. We are currently in the process of designing different decision making algorithms to determine the best path to travel.

5. EXPERIMENTAL RESULTS

We have undertaken many test runs to evaluate the control system of the AutoRoVe. With the current decision making algorithms, we are able to navigate most of the flat terrain with few large obstacles. A total of six (6) real world scenarios, ranging from zero (0) to three (3) obstacles were designed to test the AutoRoVe's ability. The autonomous command was executed in the MATLAB command prompt to test the vehicle's path planning by detecting and avoiding obstacles.

The autonomous control system was expected to acquire an image; estimate distances; obtain edge maps; define obstacles and incorporate distances to the edge map and navigate through the obstacles. The following screenshots show the analysis taken by the autonomous control system in a single obstacle scenario, as shown in Fig. 5 to Fig. 10 below.

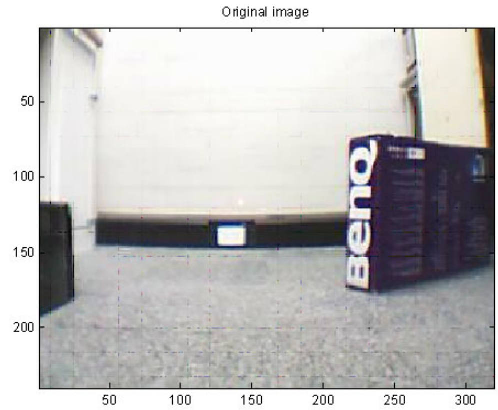

Fig. 5: Initial Vehicle View

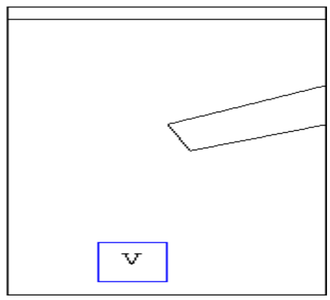

Fig. 6: Single Obstacle Scenario

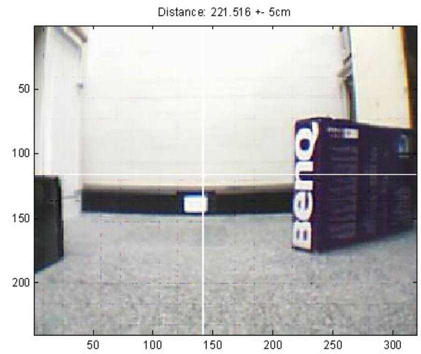

Fig. 7: Measured Distance

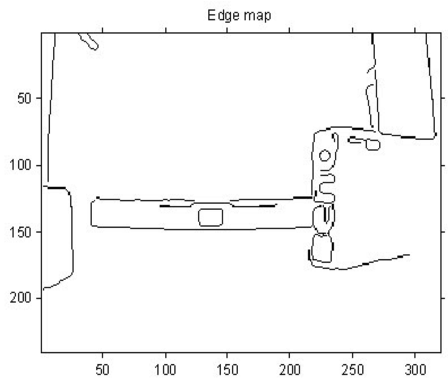

Fig. 8: Edge Map

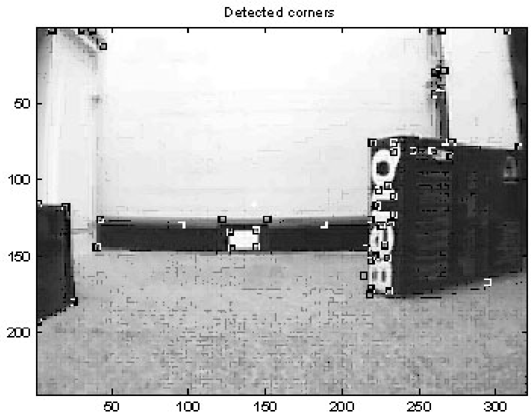

Fig. 9: Detected Corners

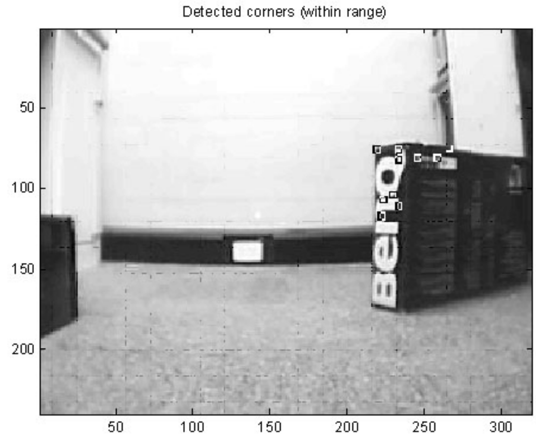

Fig. 10: Detected Corners of an Object after Being Classified as a Single Object

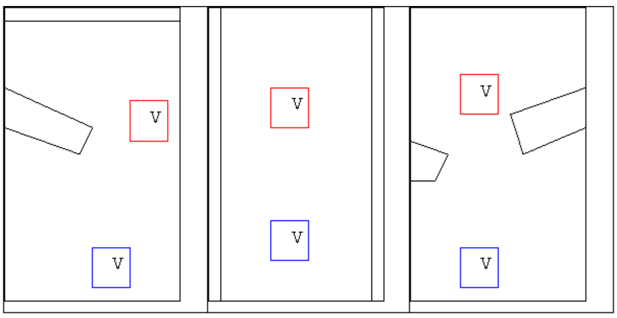

Fig. 11.1: Single Obstacle Scenario

Fig. 11.2: Corridor Navigation

Fig. 11.3: Two Obstacle Scenario

Fig. 5 shows the initial image captured by the autonomous control system. From this image, the distance between the AutoRoVe and the wall was estimated to be 221.516 ± 5cm, as shown in Fig. 7. The actual distance between the vehicle and the wall was 220cm. An edge map was also produced to assist in corner detection, as shown in Fig. 8. Fig. 9 and Fig. 10 above show the corners detected in the designed scenario. After completing the corner detection, an obstacle was found in the image with the coordinates. The camera was then observed to pan to the mid-point of the obstacle. The obstacle region was captured and the distance to the obstacle was again measured.

The distance measured between the vehicle and the mid-point of the obstacle region was 84.341 ± 5cm, where the actual distance to the obstacle is 86cm. The vehicle was then observed to spin slightly to the left and moved forward for approximately 100cm before spinning right. All the above observations could conclude that autonomous control system developed is able to plan its own traversal path without remote controlled access.

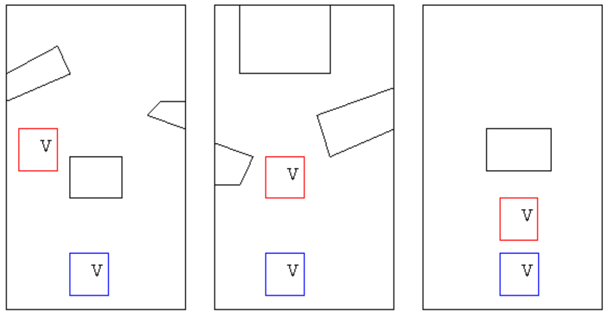

Fig. 12.1: Three Obstacle Scenario

Fig. 12.2: Three Obstacle Scenario

Fig. 12.3: Single Obstacle Scenario

The autonomous control system was further tested on different scenarios, as shown in the diagrams of Fig. 11 and Fig. 12. The blue box indicates the initial position of the vehicle whereas the red box indicates the final position of the vehicle after one execution of the autonomous command in the MATLAB command prompt.

All the above scenarios designed, as shown from Fig. 11 to Fig. 12 were the scenarios that the vehicle successfully traversed with one execution of the autonomous command. In scenarios shown in Fig. 12.2 and Fig. 12.3, the vehicle was not able to pass through any of the obstacles since the maximum distance that it could navigate was limited by the distance to the obstacle in the middle of the path.

Finally, the experiments conducted were considered as a success although the efficiency and the run-time of the algorithms could be further improved.

The software is now fine-tuned to run the autonomous command in an infinite loop instead of one execution at a time, in order to test the true ability of the autonomous control system to navigate around avoiding all the obstacles.

6. CONCLUSION

The AutoRoVe project has marked a major milestone in self-navigating capability of small robotic vehicles. The vehicle is also capable of being preprogrammed to traverse any path by making few button clicks using the simple GUI. The current version of the robotic vehicle is now getting a major upgrade with stereo vision and a retractable hoist mounted camera that will provide an aerial view of the landscape in order to make better navigational decisions.

REFERENCES

[1] Harris, T. (2005, May) "How Robots Work" (How Stuff Works). [Online]. Available: http://electronics.howstuffworks.com /robot.htm/printable

[2] Bonasso, R.P, Kortenkamp, D., and Murphy, R., "Introduction: Mobile Robots – A Proving Ground for AI", in *Artificial Intelligence and Mobile Robots*, Kortenkamp, D., Bonasso, R.P., and Murphy, R., Eds. United States of America: AAAI Press and The MIT Press, 1998, pp. 3-18.

[3] Washington, R., Golden, K., Bresina, J., Smith, D.E., Anderson, C., and Smith T, "Autonomous Rovers for Mars Exploration", in Proceedings of *IEEE Aerospace Conference*, 1999.

[4] "Autonomous Systems: Maps and Path Planning", lecture notes for WE-DINF-8524 *Autonome Systemen* (Autonomous Systems), Artificial Intelligence Laboratory, Vrije Universiteit Brussel, 2004.

[5] Bradley, J. (2001). Preliminary Concepts for a Long-Range Mars Rover Navigation System Prototype based on a Mars Global Terrain Database, Department of Computer Science, University of Calgary, Canada. [Online], Available: http://pharos.cpsc.ucalgary.ca/Dienst/Repository/2.0/Body/ncstrl.ucalgary_cs/2001-695-18/pdf

[6] "Autonomous Systems: Sensors", lecture notes for WE-DINF-8524 *Autonome Systemen* (Autonomous Systems), Artificial Intelligence Laboratory, Vrije Universiteit Brussel, 2004.

[7] "Autonomous Systems: Computer Vision", lecture notes for WE-DINF-8524 *Autonome Systemen* (Autonomous Systems), Artificial Intelligence

[8] Goldberg, S.B., Maimone, M.W., and Matthies, L., "Stereo Vision and Rover Navigation Software for Planetary Exploration" in Proceedings of *IEEE Aerospace Conference*, March 2002.