19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 December 2011
http://mssanz.org.au/modsim2011

Simulation of a mobile robot navigation system

A. Sh. Khusheef[a], G. Kothapalli[a], and M. Tolouei-Rad[a]

[a] *School of Engineering, Edith Cowan University, Western Australia*
Email: ahmedk@our.ecu.edu.au

Abstract: Mobile robots are used in various application areas including manufacturing, mining, military operations, search and rescue missions and so on. As such there is a need to model robot mobility that tracks robot system modules such as navigation system and vision based object recognition. For the navigation system it is important to locate the position of the robot in surrounding environment. Then it has to plan a path towards desired destination. The navigation system of a robot has to identify all potential obstacles in order to find a suitable path. The objective of this research is to develop a simulation system to identify difficulties facing mobile robot navigation in industrial environments, and then tackle these problems effectively. The simulation makes use of information provided by various sensors including vision, range, and force sensors. With the help of battery operated mobile robots it is possible to move objects around in any industry/manufacturing plant and thus minimize environmental impact due to carbon emissions and pollution. The use of such robots in industry also makes it safe to deal with hazardous materials.

In industry, a mobile robot deals with many tools and equipment and therefore it has to detect and recognize these objects and then track them. In this paper, the object detection and recognition is based on vision sensors and then image processing techniques. Techniques covered include Speeded Up Robust Features (SURF), template matching, and colour segmentation. If the robot detects the target in its view, it will track the target and then grasp it. However, if the object is not in the current view, the robot continues its search to find it. To make the mobile robot move in its environment, a number of basic path planning strategies have been used. In the navigation system, the robot navigates to the nearest wall (or similar obstacle) and then moves along that obstacle. If an obstacle is detected by the robot using the built-in ultrasonic range sensor, the robot will navigate around that obstacle and then continue moving along it. While the robot is self-navigating in its environment, it continues to look for the target.

The robot used in this work robot is scalable for industrial applications in mining, search and rescue missions, and so on. This robot is environmentally friendly and does not produce carbon emissions. In this paper the simulation of path planning algorithm for an autonomous robot is presented. Results of modelling the robot in a real-world industrial environment for testing the robot's navigation are also discussed.

Keywords: *Mobile robot, navigation, image processing, sensors*

1. INTRODUCTION

Industrial robots have been used widely in manufacturing setting for performing various tasks. More recently, attention has turned to add mobility to industrial robots which means the robot can perform the same tasks in different locations. In this scenario, industrial robots have to work autonomously and they must be equipped with required tools to be able to understand the environment in order to carry out its motion.

There are many types of manufacturing operations and environments for which industrial mobile robots can be used to search, find, and relocate objects or tools. In this case, mobile robot needs to be able to model its environment and identify objects. Modelling includes the process of mapping the environment based on the information of mobile robot sensors in order to determine the position of various entities, such as landmarks and/or objects. Without this mapping the mobile robot cannot find objects in the environment or plan its path to target location (Kazem, Hamad and Mozael, 2010).

There are different techniques for modelling industrial environments for the mobile robots. For instance, Ng and Braunl (2007) presented a guide tracking method in which the mobile robot is provided with a trail from starting point to the target location. The benefit of a trail is that the mobile robot reaches the target location with little autonomous navigation skills. However, the trail needs to be shaped previous to the robot navigation process. Fukazawa, et al. (2003) presented the idea of points-distribution path generation algorithm in which the robot is given a set of points that completely cover the environment. The robot in this work sought the shortest path that covered all these points. Furthermore, the robot kept looking for an object while it moves along the path and then rearranging the detected object. Another research proposed an efficient approach for modelling the search path by minimising expected time required to find the target (Sarmiento, Murrieta and Hutchinson, 2003). In this work, the mobile is equipped with efficient sensors, the environment is completely known, the object is somewhere in the environment, and the motion strategy is developed to enable the robot to quickly find the target. Jan et al. (2008) presented some optimal path planning algorithms suitable for searching a workspace in the plane image. The top image of the environment was taken and then it was divided into discrete cells. The robot was represented by a single cell so it could travel from the start location to the target location along an optimal path in the grid plain without any collisions. This method can be criticised for making use of a camera at a fixed position. In another work, Negishi et al. (2004) proposed a map generation method for mobile robot navigation in an unknown static environment. The robot was equipped with a laser range finder and two cameras were placed vertically to detect obstacles. Various techniques have been used for enabling the robot to navigate in its environment. For instance, in the work published by Abdellatif (2007) the robot was tracking by following a coloured target. In this work colour segmentation was applied to recognize the object and then the location of the target in the image was determined. In addition, a camera with three range sensors was used to detect obstacles and target distances. The camera and range sensors outputs were used as inputs for a fuzzy controller, which enabled the mobile robot to follow the object while avoiding obstacles. Abdollatif's work can be criticised for limitation on making use of a single colour for target detection. Some researchers have used environmental features as landmarks. For example, Zhichao and Birchfield (2008) explained a new algorithm which detects door features such as colour, texture, and intensity edges from the image. The extracted door information in this work was used as a landmark for indoor mobile robot navigation. Another research published by Murali and Birchfield (2008) investigated the use of corridor ceiling light for indoor robot navigation. In their work, the robot always performed straight line navigation in the centre of a corridor by keeping a ceiling light in the middle of the image; however, it restricts the motion of the robot only under the ceiling light.

This paper describes the method of modelling industrial environment (mechanical workshop) for a search mobile robot. The mobile robot is employed for searching the machines tools and then handling and transferring these equipments to target locations (delivery locations). The next sections will explain the modelling method. First, it will start with modelling environment and then it will describe the mobile robot platform. Next, control system will be described and finally the experimental results will be explained.

2. MODELLING ENVIRONMENT

The robot requires learning of certain targets in the environment and then the robot must be able to navigate between any two locations by using sensor feedback. In the environmental modelling, the mobile robot deals with three main environmental aspects: targets, tool box, and walls and/or obstacles. The mobile robot is taught to search for targets which are positioned in unknown locations and then place them in the tool box. The legged robot is used in this work therefore the process can also be performed on rough terraions.

3. THE MOBILE ROBOT

Figure 1 shows a six-legged mobile robot that is used to test the functionality of the control system. The robot has a Roboard controller which has a Vortex processor that runs at 1 GHz with 256 Mb onboard memory and only consumes 300 mA at 6 V. The Linux operation system is installed in the robot's main controller (Roboard). The robot has a gripper for grasping targets. The robot consists of 20 stepper motors; three for each leg and two for the gripper. Several sensors are equipped for detecting the environmental status around the robot.

The range sensor used is a Devantech SRF02 ultrasonic range finder. The minimum and maximum range distance that is measured by the sensor could be 20 and 180 cm, respectively. Force sensors are added to each leg and the gripper to determine the force that is applied on targets by the robot.

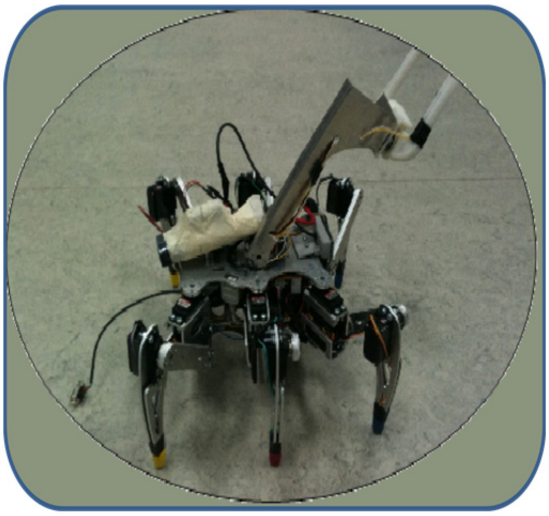

Figure 1. The legged robot.

4. CONTROL SYSTEM

A robot control system requires various sensors, including vision, range, and force sensors to obtain environmental information and convert it into digital or electrical signals. The control system converts battery energy to robot movement by using sensor signals to control and command the motions of actuators. Figure 2 shows the control process of the legged mobile robot used in this work. The following section describes the main parts of the control system.

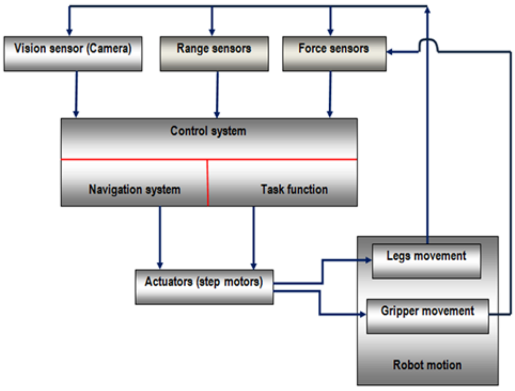

Figure 2. The control process of the mobile robot.

4.1. Navigation System

Figure 3 illustrates a flowchart describing how the navigation system works. The navigation system performs three main functions; object detection, tracking, and path planning. At the initial position, the robot performs a search for the object; if detected moves towards it and grasps it; if not detected plans a path to move to a new position in order to continue the search. Therefore, decision making is dependent on the results of image processing.

In the following sections, the main components of the navigation system will be explained. First, object detection will be described and then tracking and path planning strategies will be explained by using flowcharts provided.

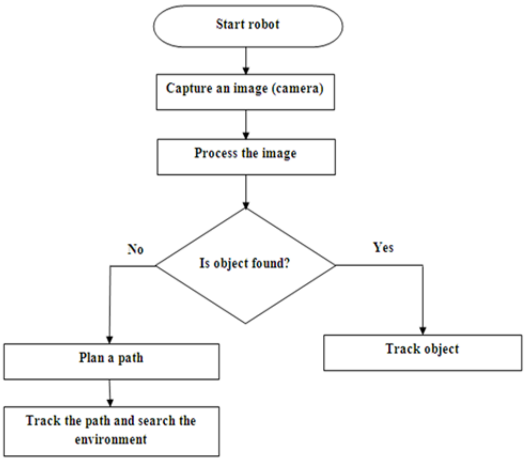

Figure 3. The navigation system flowchart.

Object Detection

A vision system is used to detect and recognise objects in the environment. Different image processing techniques are adopted and combined to achieve optimal results. In this work colour segmentation, template matching, and Speeded Up Robust Features (SURF) algorithms are employed for detection. The robot has been programmed to perform tracking once target is detected, or to continue the search otherwise. If the target is not detected in the first image taken by camera, the robot turns by 20° clockwise about its central axis and takes another image. This process is repeated until the target is detected. After a full rotation of 360° if the target has not yet been detected, a path will be planned by the navigation system to move the robot to a new location and repeat the search. Figure 4 shows a flowchart describing the working mechanism of the search and tracking in the initial position of the robot.

Tracking

When the object is detected the robot tracks it by moving towards it while keeping its image within the image plane. During tracking the robot controller continuously updates the position of the robot relative to the object by using the information coming from sensors. When the robot reaches a pre-determined distance from the target the task function is performed, that is, it grasps the object and moves it to the target location.

Path Planning

Path planning enables the robot to perform two tasks; tracking, and moving to a new position for the search. In the search mode, path planning is performed in a way that the entire search area is covered since the target position is unknown. In the proposed path planning method, the robot moves to the nearest wall (or similar obstacle) and then it turns right and continues moving along that obstacle. Any obstacle that exists in a robot path will be considered as a structure which is similar to the wall and the robot turns right and navigates along this obstacle. When the robot reaches the obstacle's boundaries, it turns left and continues moving along it. This can enable the robot to move around the obstacle. The robot uses the built-in range sensors to estimate the distance to the wall or to obstacles. When the robot is self-navigating in its environment, it continues the search. If the robot detects the object at any time, it leaves the wall, tracks, grasps, and relocates it to the target location. The robot uses the force sensors to sense the object and control the applied force by robot's gripper. Figure 5 shows the path planning flowchart.

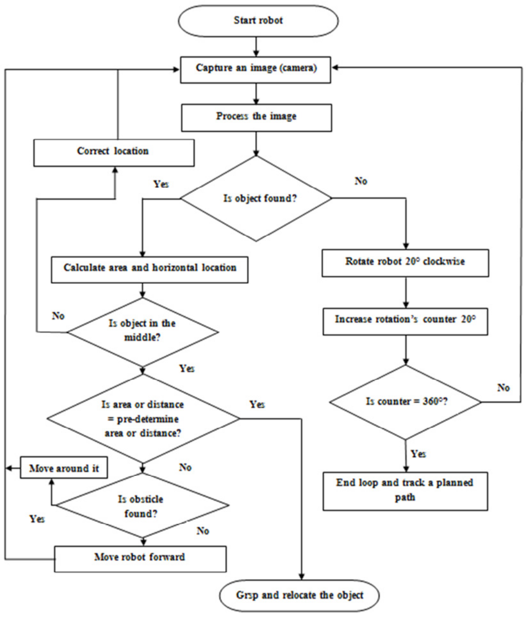
Start robot
Capture an image (camera)
Process the image
Correct location
Yes
Is object found?
No
Calculate area and horizontal location
Rotate robot 20° clockwise
No
Is object in the middle?
Increase rotation's counter 20°
Yes
Is area or distance = pre-determine area or distance?
Yes
Is counter = 360°?
No
Yes
Move around it
No
Is obsticle found?
Yes
No
End loop and track a planned path
Move robot forward
Grsp and relocate the object

Figure 4. The object searching and tracking flowchart.

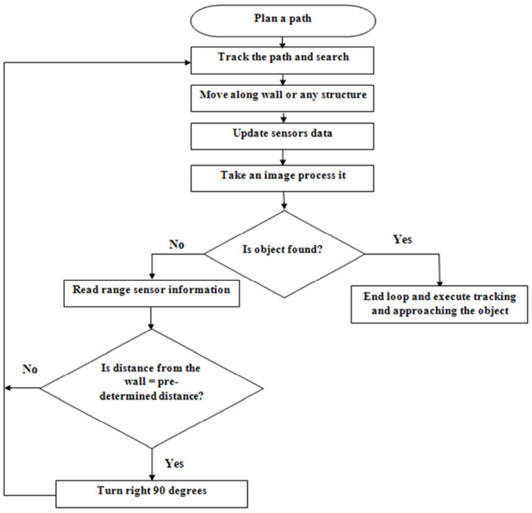
Plan a path
Track the path and search
Move along wall or any structure
Update sensors data
Take an image process it
No
Is object found?
Yes
Read range sensor information
End loop and execute tracking and approaching the object
No
Is distance from the wall = pre-determined distance?
Yes
Turn right 90 degrees

Figure 5. Path planning flowchart.

5. CASE STUDY

Experiments have been performed in two different environments in order to prove the effectiveness of the methodology developed. First, the developed program has been exercised in the laboratory and then practised in a real world environment. During the experiments, the environments were static with no dynamic obstacles. Three environmental aspects were used; a machine tool, a tools box, and walls or similar obstacles. All codes have been implemented in C++ and a number of videos and data processing libraries have been used that included OpenCV (a product of Intel Inc.) and OpenSURF (a product of Chris Evans Development). Each control system function has been tested separately and then integrated in the control program. This is shown in Figures 3, 4 and 5.

5.1. Image Processing

In this section the object detection techniques that have been used are explained. First, a colour segmentation technique has been employed for detecting the target by making use of its colour. RGB (red, green and blue colour space) and HSI (Hue, Saturation and Intensity colour space) techniques were revised and implemented for detection of the target. It was concluded that the HSI technique is less sensitive to changes in light intensities, and also it is easier to find the object by its colour. However, the output of the camera is in the RGB colour space. As a result, the colour space transformation between RGB and HSI affects real-time behaviour. Figure 6 shows colour segmentation results by using RGB colour space.

The template matching method (Utkarsh, 2010), which is a technique in image processing for finding small parts of an image that match a template image, has also been implemented and tested. Template matching is affected by object distance, which changes the target scale in the image. This problem has been solved by using different sized templates. However, it was noticed that this is an extremely slow process in real-time image processing. Figure 7 shows the template matching results.

Finally, finding the target by using the SURF technique, which is a robust image detection and description method used in computer vision systems for object recognition, has been tested and Figure 8 illustrates the result. This method has proved to be quicker than the template matching. In all the above-mentioned techniques the coordinates of the centre of the target image have been calculated and used. It has also been concluded that when an object in a single colour is used then optimal results are achieved by using the colour segmentation method. However, the template matching and SURF methods achieve better results when multi-colours are used.

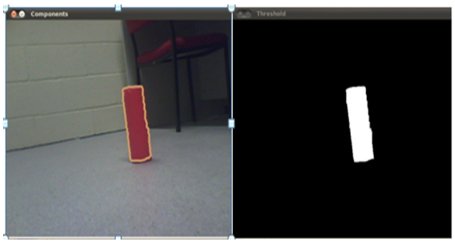

Figure 6. Object detection by the colour segmentation technique: original image is on the left and binary image is the right.

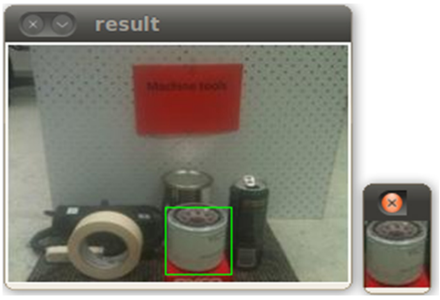

Figure 7. Object detection by template matching technique: original image is on the left and template image is on the right.

5.2. Search, Tracking, and Path Planning

The tracking codes employ the detected object information that has been extracted from the previous step as explained. The robot starts to move when the target (machine tool) is detected in the field and keeps moving towards the object. In doing so it keeps the image of the target within the image plane. When the robot reaches to the pre-determined distance from object, it grasps and puts it in the tool box. This is shown in Figure 9.

The path planning code is implemented and tested to verify the effectiveness of the method used. The robot moves along the walls or any obstacle similar to walls while searching for the object which is located in an unknown position. When the robot detects the object, it leaves the wall and moves towards it as shown in Figure 10.

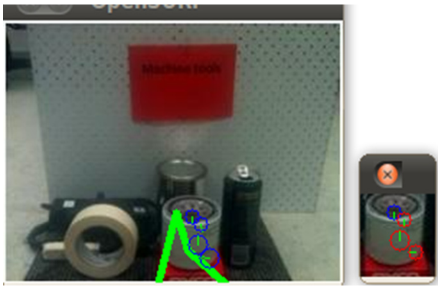

Figure 8. Object detection by the SURF technique: original image is on the left and template image is on the right.

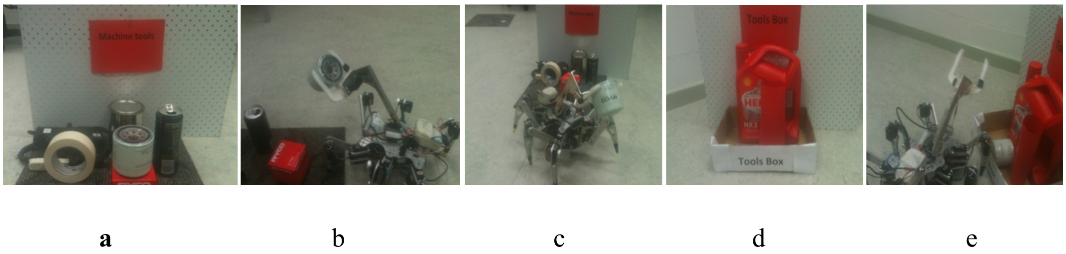

Figure 9. Object tracking, grasping and relocating.

After the successful experiment in the laboratory environment the robot has been tested in the real-world industrial environment (a car engine repair workshop). The task has been to find a pre-determined object (engine oil container) and place it on a pre-defined position. The robot proved to be able of performing the task successfully. This is shown in Figure 11.

6. CONCLUSSIONS AND FUTURE WORK

In this work the industrial environment has been modelled for the mobile robot by dividing the environment into three categories, object position, target location, and walls or similar obstacles. Different image processing techniques have been employed in order to achieve optimal results for object detection. The control system which enables the mobile robot to track or/and search the object in the environment has been implemented and tested in both laboratory and industrial environments. Experiments proved that the methodologies used and the codes developed make the robot capable of performing its tasks of finding, tracking, and relocating objects as planned. The robot can be used in real-world industrial environments to perform various find and relocate tasks as required. As a future work extension of this work, it is intended to work on an autonomous mobile robot using an artificial intelligent controller by making use of Fuzzy or neural network controllers.

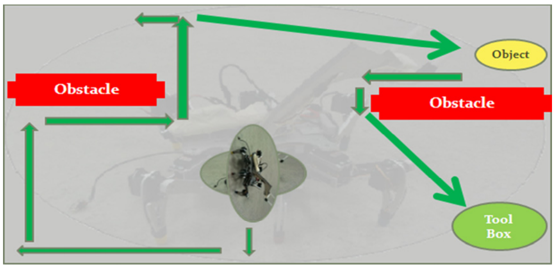

Figure 10. Searching the environment for the object.

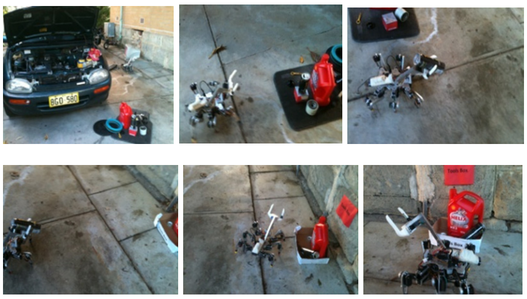

Figure 11. The robot in action in real-world environment.

REFERENCES

Abdellatif, M. (2007). A vision-based navigation control system for a mobile service robot. Paper presented at the SICE Annual Conference, Kagawa University, Japan, Sept. 17-20.

Fukazawa, Y., Trevai, C., Ota, J., Yuasa, H., Arai, T., and Asama, H. (2003). Controlling a mobile robot that searches for and rearranges objects with unknown locations and shapes. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, Oct. 27-31.

Jan, G. E., Ki Yin, C., and Parberry, I. (2008). Optimal Path Planning for Mobile Robot Navigation. *IEEE/ASME Transactions on Mechatronics*, 13(4), 451-460.

Kazem, B. I., Hamad, A. H., and Mozael, M. M. (2010). Modified vector field histogram with a neural network learning model for mobile robot path planning and obstacle avoidance. *International Journal of Advancements in Computing Technology, 2*(5), 166-173.

Murali, V. N., and Birchfield, S. T. (2008). Autonomous navigation and mapping using monocular low-resolution grayscale vision. Paper presented at the Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference, Anchorage, AK, USA, June 23-28.

Negishi, Y., Miura, J., and Shirai, Y. (2004). Mobile robot navigation in unknown environments using omnidirectional stereo and laser range finder. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka, Japan, Sept. 28 – Oct. 2.

Ng, J., and Braunl, T. (2007). Robot navigation with a guide track. Paper presented at the Robotics and Autonomous Systems, Forth International Conference on Computational Intelligence, Plamerston North, New Zealand, Nov. 28-30.

Sarmiento, A., Murrieta, R., and Hutchinson, S. A. (2003). An efficient strategy for rapidly finding an object in a polygonal world. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, USA, Oct. 27-31.

Utkarsh (2010). Available in: http://www.aishack.in/2010/01/template-matching/

Zhichao, C., and Birchfield, S. T. (2008). Visual detection of lintel-occluded doors from a single image. Paper presented at the Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference, Anchorage, AK, USA, June 23-28.