Obstacle avoidance for a non-holonomic vehicle using occupancy grids

Kane Usher
Autonomous Systems Laboratory, CSIRO ICT Centre
P.O. Box 883, Kenmore, Qld 4069, Australia
Email: Kane.Usher@csiro.au

Abstract

In this paper, we outline the strategy used for obstacle avoidance on our small, non-holonomic test vehicle, the Autonomous Tractor. This strategy relies on the fusion of data from a stereo camera, a scanning laser range-finder and the vehicle odometry to create occupancy grids which describe the traversability of the terrain in the current vicinity of the vehicle. Knowledge of the vehicle's kinematics and its response to control inputs is then used to derive obstacle free paths, if they exist, where preference is given to those commands which are 'closest' to those issued by the overarching navigation system.

1 Introduction

This paper addresses the 'local'[1] obstacle avoidance problem for non-holonomic mobile robots operating in unknown environments. Here, we consider the problem in which the obstacle avoidance vehicle controller receives demands from some higher-level, which may conflict with what is possible given the currently sensed environment. It is desirable that the modifications to the vehicle trajectory are minimal, keeping it as close as possible to the originally 'planned' path and flagging an error if this cannot be the case. A further constraint is that the method must execute in real-time.

Obstacle avoidance in general is a well studied problem for vehicles ranging from relatively simple, indoor holonomic platforms operating with minimal sensors, through to large vehicles, fitted with very complex and expensive computation and sensing suites, operating in unknown terrain. The recent DARPA Grand Challenge provides some good, and bad, examples of the current state of the art in obstacle avoidance. The solution proposed here is like many of the existing solutions — in effect, this is an applications focused paper. However, one key difference in this work is that the proposed method considers the *swept path* of the vehicle in deciding on a suitable trajectory.

[1] *Local* meaning within the vehicle's sensory perception horizon and within a finite history of these perceptions.

1.1 Paper outline

The paper is structured as follows: Section 2 overviews some of the relevant literature in obstacle avoidance; Section 3 describes the experimental platform used in this work; Section 4 outlines the occupancy grid representation used here and the creation of these grids from stereo and laser data; Section 5 outlines the obstacle avoidance strategy and describes some experimental results; while Section 6 provides some concluding remarks.

2 Literature review

The available literature on obstacle avoidance for robots and mobile robots is immense. One reason for this is that many approaches are necessarily specific to the robot in question. Early work in the area includes the research of Khatib [Khatib, 1986] who popularised the idea of using Artifical Potential Fields (APFs) to 'repel' a robot away from undesirable areas and to 'attract' the robot towards goal regions. The idea has been refined over the years and is still both useful and popular with examples of the evolution of the idea including the work of Elnagar et al. [Elnagar and Hussien, 2002] who used Maxwell's equations to produce the APFs, and Feder et al. [Feder and Slotine, 1997] who used the Laplace equations.

Problems with APFs include 'potential wells' in which the robot can get 'stuck'; at least for 'local' obstacle avoidance, the problem is not unique to APF methods. Methods exist to create APFs without these 'wells' (see e.g [Rimon and Koditschek, 1992]), however these methods, along with the methods relying on ideas from electro-magnetics and fluid mechanics, are highly computationally intensive and require global knowledge of the structure of the environment. Some APF methods have also been developed to deal with non-holonomic systems (see e.g. [Tanner *et al.*, 2001]) but these meth-

ods are usually computationally intense and difficult to adapt for systems with significant dynamics.

One approach to overcome some of the shortcomings of APFs includes the Vector Field Histogram (VFH) method developed by Koren and Borenstein [Koren and Borenstein, 1991]. This method was originally targetted at robots equipped with ultra-sonic sensors, addressing some of the issues associated with this mode of sensing. It also addressed the 'narrow passageway' oscillations found with some APF method implementations. Essentially, this approach builds up a polar histogram of the range and bearing structure of the surrounding environment, selecting the path which has the least potential for a collision within the histogram. The VFH method has since been extended to what is termed VFH+ [Ulrich and Borenstein, 1998] which reduces the parameter tuning required of VFH by explicitly compensating for the width of the robot. However, this approach does not take into account constraints such as non-holonomy. Further work includes VFH* [Ulrich and Borenstein, 2000] which provides a look-ahead mechanism, mitigating some of the local minima issues.

The work of our own group has included a reactive navigation method for a Load Haul Dump Vehicle [Roberts *et al.*, 2000], operating at speeds of up to 30 km/h through a network of mine tunnels. This system used the idea of active contours to create a 'snake' which kept the vehicle from hitting the tunnel walls, and could be influenced to navigate tunnel forking configurations as desired. This work has since been adapted for use in road following (see [Usher *et al.*, 2004]) which is similar to the tunnel environment but this method has not been adapted for open-space use.

The Dynamic Window Approach (DWA) developed by Fox et al. [Fox *et al.*, 1997] can deal with non-holonomic and dynamic aspects of robot characteristics. The method operates by evaluating a set of possible robot trajectories using some cost associated with, for example, minimum distance to an obstacle, minimum divergence from original trajectory etc. Most implementations and extensions to this method deal with point or circular robots (see e.g. [Ögren and Leonard, 2005] for a recent example in which the idea is developed in a model-based control framework).

The DARPA Grand Challenge elicited perhaps some of the most sophisticated, and high speed, applications of obstacle avoidance behaviours. Stanley, the eventual winner of the Challenge, used a method which is similar in principle to the DWA approach, analysing a series of trajectories against some 'cost'. The trajectories spanned the space of 'swerve' and 'nudge' manouvres which amounted to trajectories with maximum and minimum lateral accelerations respectively. However, Stanley, like most of the other Grand Challenge entries was both sensor and computer rich, allowing for the application of relatively sophisticated methods at a rate sufficient for the operational speeds encountered. The constraints of the road, such as they were at times, also helped shape the required trajectories.

Figure 1: The Autonomous Tractor highlighting the sensors used in these experiments.

The work presented in this paper most closely resembles the Dynamic Window Approach, however we deal with a non-circular robot with significant motion constraints, and we deal with the motion in an image space. Unlike, the Grand Challenge robots, we have relatively limited computational resources, and operate at a much lower speed (< 12 km/h). Additionally, we consider the *swept path* of the vehicle in evaluating trajectories.

3 Experimental platform

The CSIRO Autonomous Tractor (AT), as shown in Figure 1, is an Ackerman steered vehicle with a top forward speed of approximately 12 km/h. The platform is a ride-on mower which has been retro-fitted with an array of actuators, sensors, and a computer system enabling the implementation and testing of control and navigation algorithms. The vehicle's design is such that it can be operated in three modes: *manual*, *remote*, or *automatic*. Manual operation is the AT's original mode of operation in which the vehicle is driven by an operator. Remote mode allows a user to control the vehicle from a hand-held radio-transmitter. Automatic mode allows the vehicle to be controlled by the on-board, or a networked, computer. For full details of the vehicle's design, refer to [Usher, 2005].

In the experiments cited here, the AT is controlled via a joystick, from which the demands are modified by the obstacle avoidance software before being sent to the vehicle (similar to the wheelchair guidance work of [Bell *et al.*, 1994]). In these experiments, the sensors used are:

1. a front-mounted Videre stereo camera,

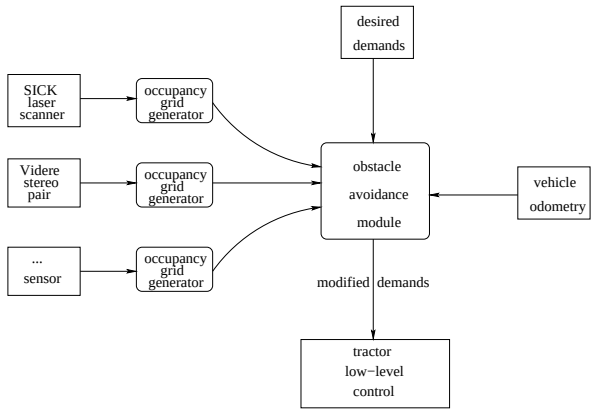


Figure 2: Abstracted view of the software and sensors used in these experiments.

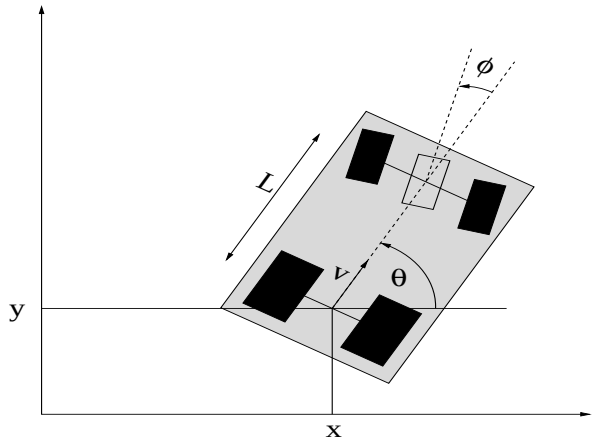


Figure 3: System kinematics.

2. a front mounted SICK Laser Measuring System (LMS), and
3. the vehicle odometry.

Figure 1 shows the mounting position of the stereo camera and the laser.

Figure 2 shows an abstracted view of how the obstacle avoidance system operates. The communication between the various modules actually occurs using the Dynamic Data Exchange system [Corke *et al.*, 2004]. Of course, the demands could be issued from any source; here a joystick is used for experimental and illustrative purposes.

As noted earlier, the AT is Ackerman steered (i.e. it is a car-like vehicle) meaning that its kinematics are non-holonomic. A non-holonomic system is difficult to control as its configuration is dependent on the path taken to reach that particular configuration. Here, we use knowledge of the kinematics, together with the system dynamics and constraints, to enable simulation and prediction of possible trajectories of the vehicle given a particular combination of control inputs.

3.1 Kinematics

Referring to Figure 3, the kinematics in Cartesian space of the AT are:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= v\frac{\tan\phi}{L} \end{aligned} \tag{1}$$

where v is the vehicle's forward velocity (measured at the centre axle of the rear wheels), L is the vehicle's length, ϕ is the steering angle, the point (x, y) refers to the centre of the rear axle, and θ is the vehicle's orientation.

3.2 Dynamics

Many control algorithms for car-like vehicles fail to consider the realities of implementation on a vehicle, for example steering angle limitations and the dynamics of the actuators which drive the vehicle. In this work, models of the vehicle's dynamic response to demands are used, including non-linear effects such as input saturation.

Steering

An approximate model of the steering dynamics was experimentally identified from the response of the AT's steering loop to unit step changes in desired steering angle (provided from the computer). The response was determined to be approximately second-order of the form (in the Laplace domain):

$$\frac{\phi(s)}{\phi^*(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{2}$$

The parameters ω_n and ζ where found to vary due to the complexity of the interactions between the terrain and the wheels on different surfaces and also varied with the vehicle's translational speed. For the purposes of these experiments, parameter values of $\omega = 0.72$ and $\zeta = 0.8$ were found to model the system adequately. The steering angle is also limited to $\pm 0.523\,\text{rad}$ ($\pm 30°$).

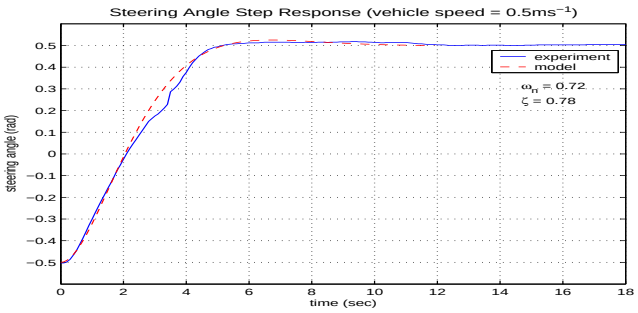
Figure 4(a) shows a plot of the actual and modelled response of the vehicle to a unit step input.

Velocity

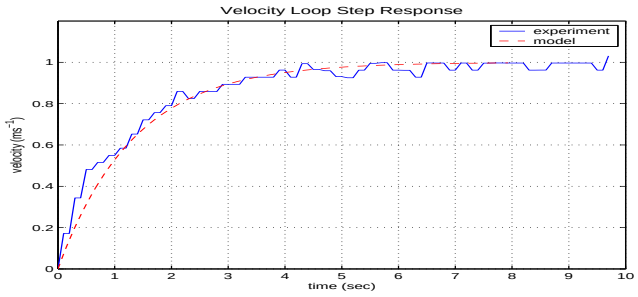
The velocity loop was empirically determined to have a first-order response which is represented in the Laplace domain as:

$$\frac{v(s)}{v^*(s)} = \frac{K_v}{\tau_v s + 1} \tag{3}$$

Figure 4(b) illustrates the AT's response to a unit step change in velocity while travelling on level ground (concrete). Also shown is the response of the first-order model where $K_v = 1$ and $\tau_v = 1.33$. Note the 'quantisation' of the experimental velocity measurements. The



(a) Step response of the steering loop.



(b) Unit step response of the speed loop.

Figure 4: Step response of the steering and velocity loops. Model data is plotted as a dashed line and experimental data is plotted as a solid line.

parameters (K_v and τ_v) will again vary slightly due to differences in the velocity response on sloping terrain, different surfaces and under different loading conditions. Additionally, the forward speed of the AT is limited to 3 m/s while the reverse speed is limited to 1.5 m/s.

3.3 Trajectory simulation

The kinematic and dynamic models developed earlier can be used to predict the motion of the vehicle given its current pose and the control inputs. Figure 5 shows a comparison of the real position of the vehicle (measured using the vehicle odometry which is relatively drift-free over the short distance considered here) against the position predicted by the system model. As will be described later, this ability to predict the vehicle's path can be used to 'test' particular trajectories for potential collisions.

4 Occupancy Grids

This section describes the creation of occupancy grids from the stereo and laser data. In all cases, the occupancy grid is referenced to the centre of the rear axle on the vehicle with the coordinate system shown in Figure 6.

4.1 Stereo-based

The stereo camera is from 'Videre design' (http://www.videredesign.com/index.htm). It pro-

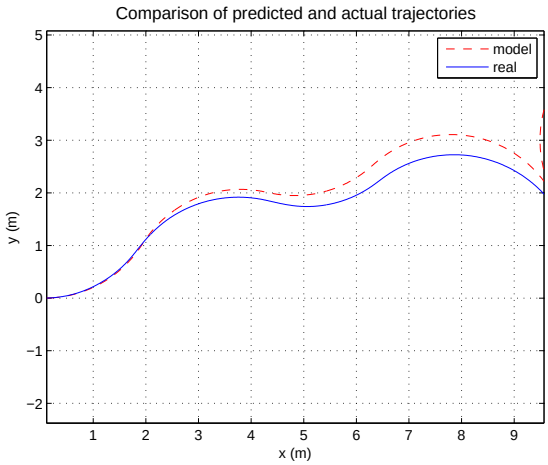


Figure 5: Real and simulated response of the vehicle to a series of control inputs for a trajectory similar to a 'swerve' around an obstacle. Model data is plotted as a dashed line and experimental data is plotted as a solid line.

Figure 6: All occupancy grids are referenced to the centre of the rear axle with the coordinate system shown. Sensor data is transformed to this coordinate frame.

vides relatively high frame rate stereo data in the form of both disparity maps and point cloud data. An image from the left camera is shown in Figure 7(a), the corresponding disparity map and point cloud are shown in Figures 7(b) and 7(c) respectively.

To form occupancy grids from the stereo data, two methods were developed. The first method relies on finding the dominant plane in the point cloud image by using a Least Median Squares (LmedS) optimisation. The second method looks at local gradients in the point cloud data. Both methods operate in real-time (which is only required to be around 5 to 10 Hz for the speeds at which the AT operates). In fact, the system is designed such that it can accept output from both methods, feeding these occupancy grids into the 'obstacle avoidance' module as illustrated in Figure 2. Each of the methods is described in more detail below.

LmedS method

The LmedS method uses a non-linear optimisation method to find the dominant plane in the point cloud data and then uses this plane to build the occupancy grid based upon the distance of each point in the cloud from the dominant plane. The LmedS method of optimisation was chosen over linear methods (such as a Least Mean Squares) due to its superior outlier rejection properties — in fact the dominant plane need form as little as 50% of the image using this approach. Clearly this is an important property if navigation is to occur in realistic environments.

Working in Cartesian space, a plane can be described by a set of four parameters p_i:

$$p_1 x + p_2 y + p_3 z + p_4 \quad = \quad 0 \tag{4}$$

In the LmedS minimisation, the left hand side of Equation 4 is evaluated at each point x_i, y_i, z_i in the point cloud and the result e_i is stored in an error vector

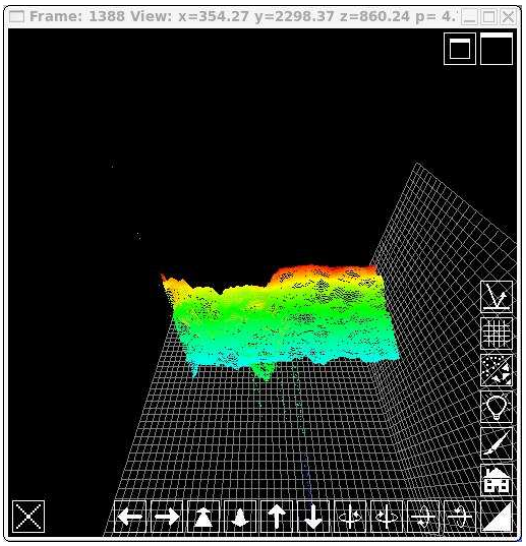
$$e_i \quad = \quad p_1 x_i + p_2 y_i + p_3 z_i + p_4 \tag{5}$$

where e_i is in fact the distance of the point (x_i, y_i, z_i) from the plane. It is the median value of this error vector which is used to drive the minimisation which here is performed using a Nelder-Mead simplex search [A. and Mead, 1965] (faster methods exist, this method is used for convenience). Figure 8 shows an example of the stereo point cloud data together with the plane fitted to the data.

Having found the dominant plane, the viewpoint height, roll and tilt with respect to the plane can be estimated. This can be done by recognizing that the parameters (p_1, p_2, p_3) in Equation 4 form a normal vector to the plane, whilst the distance d from the origin (the viewpoint) to the plane can be found using the following

(a) Example image from the left camera of the stereo pair.

(b) Disparity map from the same scene as depicted in Figure 7(a).



(c) Point cloud from the same scene as depicted in Figure 7(a).

Figure 7: An example image and the corresponding disparity map and point cloud.

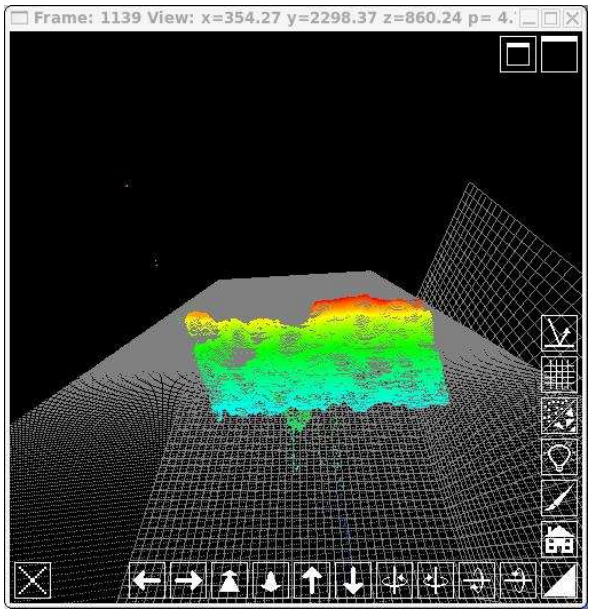


Figure 8: Example of the stereo point cloud data showing the dominant plane fitted to the data.

equation:

$$d = \frac{p_4}{\sqrt{p_1^2 + p_2^2 + p_3^2}} \tag{6}$$

Comparison of the calculated viewpoint values to the 'expected' values (from knowledge of the camera mounting geometry with respect to the ground-plane) can determine whether the discovered plane is in fact the ground-plane. If these values are in agreement, we can safely assume that the plane found is the ground-plane and proceed to create the occupancy grid. Of course some tolerance is required here as one of the advantages of the LmedS method is that it can adapt to changes in terrain slope (provided that at least 50% of the point cloud data lies on the dominant plane).

If the tolerance on the camera geometry check is not met the occupancy grid is created on the presumption that the 'expected' camera geometry is accurate, and an error is flagged informing the obstacle avoidance module that the information from this mode of sensing is likely to be unreliable.

This course of action is not unreasonable; consider a scene in which the vehicle is confronted with a vertical wall which forms greater than 50% of the image. In this case, the occupancy grid will still be populated, identifying the wall as a barrier, but the information is not as reliable as it could be and other modes of sensing should be more heavily weighted.

In any case, the occupancy grid can now be populated using the idea that points not on the ground-plane (at least within a tolerance) must belong to an obstacle. That is, we can calculate the distance of each point from the ground-plane using Equation 5. If this distance exceeds a threshold, then the point contributes to the evidence that the ground-plane cell to which the point belongs to contains an obstacle. Otherwise, if the distance is below the threshold, the point contributes to the evidence that the cell is free. Figure 9 illustrates an example occupancy grid generated using this method. In this figure, an 'empty' cell is represented by black, an 'occupied' cell is represented by white, and 'unknown' cells are represented by grey.

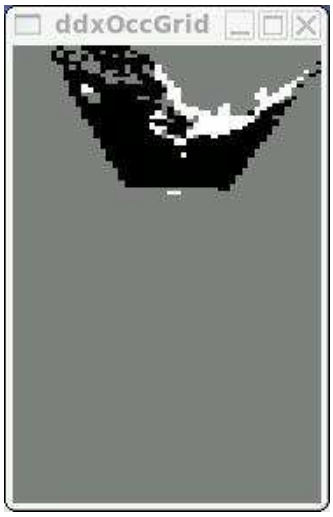


Figure 9: Example of an occupancy grid created from the stereo data using the LmedS method.

Gradient based method

This method relies on the idea that local gradients in the point cloud data can be used to determine whether a particular cell is occupied by an obstacle (more accurately, to determine whether the cell is 'traversable'). The method relies on knowledge of the camera geometry with respect to the ground-plane. The method proceeds as follows:

1. Transform the point cloud data to a coordinate system aligned with the ground-plane, as shown in Figure 6.
2. Traverse the point cloud list and at each cell find the 'median' height from the ground-plane. This essentially creates a height-encoded occupancy grid.
3. Calculate the gradients at each cell (e.g. using a Sobel image kernel) in the height-encoded occupancy grid and assign a traversability value based on a gradient threshold (empirically identified).

Figure 10 illustrates the results of this process. Not surprisingly for this simple scene the results are fairly similar to those of the LmedS method. One failure mode is

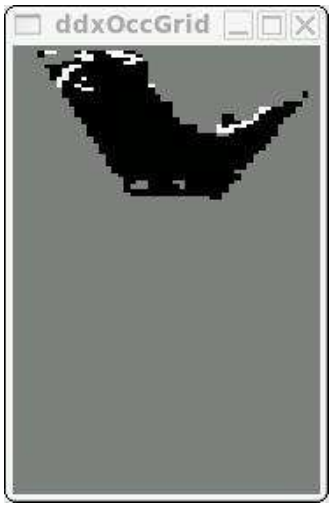


Figure 10: Example of an occupancy grid created from the stereo data using the gradient method.

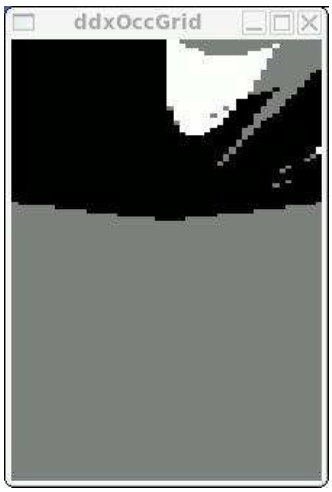


Figure 11: Example of an occupancy grid created from the LMS.

when the camera geometry with respect to the ground-plane is in error; the coordinate transformation will then be incorrect which could lead to errors in determining the traversability of a cell. Such a situation could occur if the vehicle encountered a steep hill.

Discussion

The methods are somewhat complimentary; the LmedS method requires at least 50% of the points to lie on the dominant plane — the gradient method does not; the LmedS method can adapt to changes in the ground-plane (i.e. if the vehicle encounters a hill, the method does not fail) — the gradient method can fail because the coordinate transformation will be incorrect. For the gradient method, the latter issue can be overcome by using the camera geometry as identified in the LmedS method. This leaves the following failure mode; the vehicle encounters a hill with less than 50% of the point cloud data lying on the dominant plane (for which the dominant plane can't be identified and the camera geometry is incorrect). In field robotics this is not a particularly unlikely scenario and the failure must be flagged. Fortunately, there is enough information to flag the error in this case.

4.2 LMS based

Occupancy grids are created using the LMS in the standard ray-tracing type manner which essentially walks along each scan line to determine the occupancy of the cells in the ray's path. Figure 11 illustrates an occupancy grid constructed from the front-mounted LMS.

Currently, the system is set up to scan in a horizontal plane. Information from the LMS is much more reliable than from the cameras and the range of this sensor is also better. However, this set-up means that obstacles outside this scanning plane are missed, for example spotting 'overhanging' obstacles is difficult with the LMS. Hence, the grids constructed from the stereo data provide somewhat complementary information. Future work will look at tilting the laser down at a slight angle in order to build up local terrain maps.

4.3 Occupancy grid evolution

With knowledge of the vehicle's relative motion, the occupancy grids can be used to build up a map of the terrain that the vehicle has encountered but which may have left the particular sensor's field of view. Of course, with no other means of localisation, the relative motion estimates will begin to drift so the 'historical' information contained in these evolved occupancy grids needs to be gracefully degraded over time. The occupancy grid is only maintained in a local region around the robot (the size of the local map is adjustable – for these experiments the map-size was approximately 10×10 m with a resolution of 0.1 m) — once an obstacle drops off of the map it is 'forgotten'.

Drift on odometry is certainly an important factor in maintaining the map, particularly for the section of the map that is out of the field of view of the sensors, that is, behind the vehicle. When reversing, the information used to determine a trajectory may be somewhat 'out-of-date', however, this information is better than using no information at all. Using a more sophisticated method for localisation over odometry would improve the accuracy of this local map but still would not account for the fact that there is currently no sensing towards the rear of the vehicle.

4.4 Fusion

Occupancy grids offer a natural means of fusing data from disparate sources. Here, the occupancy grids are all referenced to the centre of the rear axle of the vehicle. A simple weighting scheme, based on the reliability of the data, is used to create a new occupancy grid from the original grids. At each cell, the cell's value is assigned as:

$$O(i,j) = \sum_{k=1}^{n} w_k O_k(i,j) \quad (7)$$

where i, j is the index to the occupancy grid cell, O_k is the k^{th} occupancy grid, w_k is the weighting for the k^{th} occupancy grid and n is the number of occupancy grids.

This is a rather simplistic means of fusing the occupancy grids but it has been adequate for the purposes of this work. Future work will look at the more sophisticated methods of occupancy grid fusion.

After fusion, the occupancy grid is then thresholded to provide a traversability map, that is a map that specifies whether a cell is occupied or free.

5 Obstacle avoidance

Given a valid traversability map, and a means of simulating the motion of the vehicle, potential trajectories of the vehicle can now be assessed. The process used here is to take the original vehicle demand, and using knowledge of the vehicle's footprint and the earlier presented vehicle motion models, calculate the swept path of the vehicle along this trajectory. This swept path is then checked against the traversability map. If this trajectory is collision free, then the commands are passed on to the vehicle. If not, then a search for a feasible trajectory is initiated. This search involves predicting the swept path of the vehicle across a discretised range of the available steering angles (here 5° intervals in the available steering angle range of ±30°) at the current speed, and testing this swept path against the traversability map. If a valid trajectory is still not available, then the demanded speed of the vehicle is recursively reduced and the process is repeated until a valid trajectory is found, or motion in the current direction is no longer possible.

5.1 Collision checking

Collision checking is performed by mapping the vehicle's shape onto a further occupancy grid and recursively transforming this shape along the predicted trajectory — this provides the swept path of the vehicle. Figure 12 illustrates an example of this process, where the vehicle's shape at the start location has been highlighted for clarity. This occupancy grid is essentially 'AND'ed with the traversability map, identifying potential collision points for that particular trajectory.

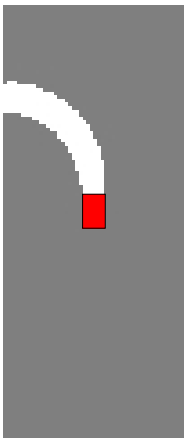


Figure 12: Example of an occupancy grid created from the transformation of the vehicle's shape to the poses contained in the predicted trajectory.

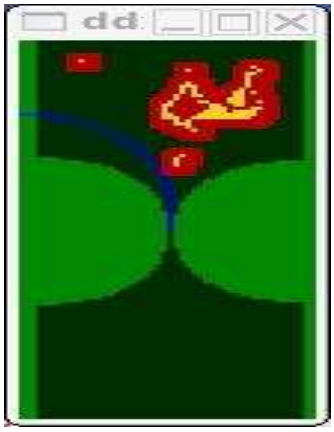


Figure 13: Example of an avoidance trajectory around the obstacles in the grid.

5.2 Experiments

An example of an obstacle avoidance trajectory is shown in Figure 13. As can be seen, this trajectory navigates the vehicle around the obstacles in the grid. For a more comprehensive illustration of the effectiveness of this method please refer to Video 1 [Usher, 2006] which accompanies this paper.

6 Conclusions

This paper has described an application of obstacle avoidance to a non-holonomic vehicle using occupancy grids and an approach similar to the Dynamic Window Approach. Notably, the method presented here deals with the swept path of the vehicle and operates in real-time. The system is designed to accept occupancy grids from a variety of sources. Here, occupancy grids from three different sources were described: the first grid was

created by finding the ground-plane from a dense stereo point cloud and then looking for items which were outside of this plane; the second method again used the stereo point cloud, this time using the gradients within the cloud to identify traversable areas; the third method used data from an LMS scanning in a horizontal plane in front of the vehicle.

In combination, these occupancy grids, together with the obstacle avoidance strategy were shown to ensure that the vehicle maintained collision free paths and came to a halt if this was not possible.

Acknowledgements

This work was funded by the CSIRO ROVER project. The technical support of the CSIRO ICT Centre Robotics Team — Stephen Brosnan, Graeme Winstanley, Jonathan Roberts, Cedric Pradelier, Pavan Sikka, Elliot Duff, Les Overs, and Polly Alexander — is gratefully acknowledged.

References

[A. and Mead, 1965] Nelder J. A. and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[Bell *et al.*, 1994] David A. Bell, Johann Borenstein, Simon P. Levine, Yorem Koren, and Lincoln Jaros. An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance. In *International Conference on Robotics and Automation*, pages 2012–2017, San Diego, California, USA, 1994. IEEE.

[Corke *et al.*, 2004] Peter Corke, Pavan Sikka, Jonathan Roberts, and Elliot Duff. Ddx: A distributed architecture for robot control. In *Proceedings of the Australasian Conferenc on Robotics and Automation*, Brisbane, Australia, December 2004.

[Elnagar and Hussien, 2002] A. Elnagar and A. M. Hussien. Motion planning using Maxwell's equations. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2347–2352, Lausanne, Switzerland, 2002.

[Feder and Slotine, 1997] Hans Jacob S. Feder and Jean-Jacques E. Slotine. Real-time path planning using harmonic potentials in dynamic environments. In *International Conference on Robotics and Automation*, pages 874–881, Albuqueque, New Mexico, USA, April 1997. IEEE.

[Fox *et al.*, 1997] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, v(1), 1997.

[Khatib, 1986] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[Koren and Borenstein, 1991] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *International Conference on Robotics and Automation*, pages 1398–1404, Sacramento, California, USA, April 1991. IEEE.

[Ögren and Leonard, 2005] Petter Ögren and Naomi Ehrich Leonard. A convergent Dynamic Window Approach to obstacle avoidance. *IEEE Transactions on Robotics*, 2005.

[Rimon and Koditschek, 1992] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.

[Roberts *et al.*, 2000] Jonathan M. Roberts, Elliot S. Duff, Peter I. Corke, Pavan Sikka, Graeme J. Winstanley, and Jock Cunningham. Autonomous control of underground mining vehicles using reactive navigation. In *International Conference on Robotics and Automation*, pages 3790–3795, San Francisco, California, 2000. IEEE.

[Tanner *et al.*, 2001] Herbert G. Tanner, Savvas Loizou, and Kostas J. Kyriakopoulos. Nonholonomic stabilization with collision avoidance for mobile robots. In *Proceedings of the 2001 International Conference on Intelligent Robots and Systems*, pages 1220–1225, Maui, Hawaii, USA, October 2001. IEEE/RSJ.

[Ulrich and Borenstein, 1998] I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. In *International Conference on Robotics and Automation*, pages 1572–1577, Leuven, Belgium, 1998.

[Ulrich and Borenstein, 2000] Iwan Ulrich and Johann Borenstein. Vfh*: Local obstacle avoidance with look-ahead verification. In *International Conference on Robotics and Automation*, pages 2505–2511, San Francisco, California, USA, 2000. IEEE.

[Usher *et al.*, 2004] Kane Usher, Jonathan Roberts, Peter Corke, and Elliot Duff. Vision-based navigational competencies for a car-like vehicle. In *Preprints Int. Symp. Experimental Robotics*, Singapore, June 2004.

[Usher, 2005] Kane Usher. *Visual homing for a car-like vehicle*. PhD thesis, Queensland University of Technology, Brisbane, Australia, June 2005.

[Usher, 2006] Kane Usher. Video 1 — CSIRO Autonomous Tractor: Obstacle avoidance. `http://www.cat.csiro.au/ict/download/acra/usher.mpg4`, 2006.