An Autonomous Terrain Coverage and Marking System for Humanitarian Landmine Clearance

Thomas Lowe[1], **Paulo Borges**[1], **Serge Lichman**[1], **David Haddon**[1], **James Brett**[1], **Matt Wildie**[1], **Paul Cronin**[2]
[1]Robotics and Autonomous Systems Group, Data61, CSIRO, Brisbane, Australia
[2]Molten Labs, Sydney, Australia

Abstract

Landmine detection is one of the most compelling cases for autonomous robotics. The risks involved for human workers in charge of identifying and clearing the devices are extremely high. A challenge around the detection and clearance process is how to indicate that an area has been cleared and is then safe. The community and workers around a minefield need a direct visual indication of scanned and unscanned regions.

We present an autonomous ground vehicle that performs field coverage and autonomously deposits white and red painted lines (these colours follow a United Nations convention) to guide the vehicle in scanning the minefield while also presenting a clear marker of scanned (white) and unscanned (red) areas, which is valid even on incomplete scans. The system consists of several modules which include: a novel line detection system coupled with a visual-servoing mechanism to follow deposited lines, a state machine to indicate what type of lines should be painted, and control modules for the autonomous vehicle. These painted lines take the place of red and white painted sticks or stones that have been used for decades in demarking these hazardous areas. We present experimental results illustrating the applicability of the system.

1 Introduction

Landmines are one of the most afflicting consequences of post-war in modern times. Although there is a global eeffort (the Mine Ban Treaty) to reduce their impact, it is very difficult to detect and remove these device especially if they are located in unstructured environments or near crop fields. Although it is impossible to know the exact number, according to the United Nation statistics and the International Campaign to Ban Landmines (ICBL), it is estimated that more than 50 million landmines exist in the ground in more than 60 countries, causing more than 10,000 deaths and 20,000 injuries every year [lan, 2017]. The cost of clearing a mine is very high, particularly to poor countries, which suffer most from the presence of mines. The clearing process is usually performed manually, which is time consuming and extremely dangerous. For those reasons, there is ongoing research for demining robots to be used in the detection and deactivation of those device.

The method for landmine removal depends on factors such as terrain, landmine distribution density, soil density, and type of vegetation. A United Nations recommendation, however, is that in the scanning process the fields are marked to indicate areas that $h$ave been cleared versus areas that have $n$ot been cleared. For this indication, markers of white and red colour are used, where the white side represents 'safe' and the red side represents 'unknown' (or danger). Traditionally, coloured rocks, sticks or paint is used to represent the scanned areas. It is very hard, though, to incorporate these marking strategies in most mobile robot demining robotic solutions.

We propose a system that uses a small car-size robotic vehicle as the platform to carry a landmine detection and deactivation device. The robot is equipped with a 'line marking' system that autonomously (and coherently) deposits red and white paint on the field based on areas that have been covered or not (Figure 3). The robot navigates with the goal of performing full coverage of a field and uses its own deposited lines as a guide for precise coverage. To follow the lines, a novel line detection strategy is proposed, which is suitable for irregular and 'noisy' terrain, and built as a visual-servoing module. A state machine determines whether the vehicle should deposit white or red paint on the ground. The following components form the full navigation and marking system:

1. The general area of interest is determined by the user through a bounding polygon in a graphical user

interface (GUI) based on satellite image maps;

2. In the field the robot then uses a combination of visual servoing with wheel odometry for odometric localisation;
3. As the robot moves, it paints the ground enabling/disabling the painting mechanism;
4. The mission is completed when the whole area has been covered and the field is marked appropriately.

Although the landmine detection itself is not part of this paper (we focus here on the robotic component - the detection device is built by a partner organisation), our goal is to illustrate that this new design is a practical alternative for efficient landmine clearance. In our current implementation the maturity of our system is suitable for the primary case of relatively flat fields without obstacles. We intend to explore the more challenging cases of minefields with obstacles and unusual field shapes in future iterations.

2 Literature Review

There exists a number of strategies for landmine detection. A significant amount of research effort has been directed towards sensor technology, which includes electrical impedance sensors, acoustic sensors, electromagnetic induction sensors, radiometry, radar, infrared and microwave sensors [Trevelyan, 1997; Won *et al.*, 2001] Robotic technology aims to incorporate one of these sensing techniques in an autonomous or remote controlled robot, in order to remove the need for humans to closely inspect the field.

Legged robots have been one of the main alternatives for demining robots due to their potential ability to navigate through uneven terrain. Earlier works [Gonzalez de Santos* *et al.*, 2005; Huang and Nonami, 2003; Estremera *et al.*, 2010] focused mostly on defining efficient designs and gait patterns. One of the disadvantages of legged platforms are their less energy-efficient locomotion and complex control.

The Gryphon robot [Fukushima *et al.*, 2005] is a four-wheeled vehicle which also performs scanning manipulation. It is design as an all terrain vehicle. The platform paints marks on the ground to indicate the locations of the mines. The Ares robot [Santana *et al.*, 2007] consists of four independently steered bicycle wheels. The design is supposed to be cost efficient and is able to use four different configurations (Ackerman, turning point, omnidirectional, lateral mode), depending on the most suitable locomotion type. The TRIDEM platform [Baudoin and others, 2008] is a three wheel robot each with each wheel having an independent driving and steering system. The vehicle is remote controlled by an operator based on the images streamed by the camera on the robot. The FSR Husky mobile wheeled robot was developed to assist humans with landmine detection [Portugal *et al.*, 2014] using a Clearpath Husky A200 robot base. A ground vehicle was developed for landmine detection based on Teodor platform [De Cubber *et al.*, 2014]. A three-wheeled mobile robot was designed for demining purposes without any suspensions system [Doroftei *et al.*, 2014] while a modified agricultural machine for remotely controlled unmanned mine-clearing robot was designed by Hemapala et al [Hemapala *et al.*, 2009]. The MIDERS-1 and 2 robots use a caterpillar chain based mobile platform [Suh *et al.*, 2014], presenting good ability to navigate through rough terrain.

A more complete review of a number of platforms is presented by Marques et al [Marques *et al.*, 2016]. All of the designs above present advantages and drawbacks regarding navigation and detection ability, but none addresses the marking of covered areas, which is an essential (and mandatory, according to official United Nations guidelines) part of the demining process. The system proposed in this paper can potentially be adapted to many of the robots above.

3 System Description

In this section we describe the main components of the system. The vehicle used in our development for the terrain navigation and terrain marking is shown in Figure 1. The image illustrates the arms used for spray-painting the ground, the cameras used to detect and follow the painted lines and the 3D localisation system for autonomous navigation.

The overall goal of the system is to cover a given area A as efficiently as possible while not missing any regions of A. An area is considered 'covered' if the vehicle traverses over it while scanning the ground with a low-resolution magnetic detector mounted in the front of the vehicle, as illustrated in Figure 2. As discussed in Section 1 this paper focuses on the navigation/coverage/marking components and the landmine detection is not part of this research, hence the landmine detector is not mounted on the vehicle for our experiments. In the following we will discuss:

- The choice of the driving pattern for field coverage.
- The design of the camera and spraying systems.
- The state-machine used for action control.
- The ground line detection system.
- The design of the forward, backwards and manoeuvring motion.

3.1 Choice of Driving Pattern

The area of interest is scanned in a 'comb' pattern whereby the vehicle drives forwards to the far edge of

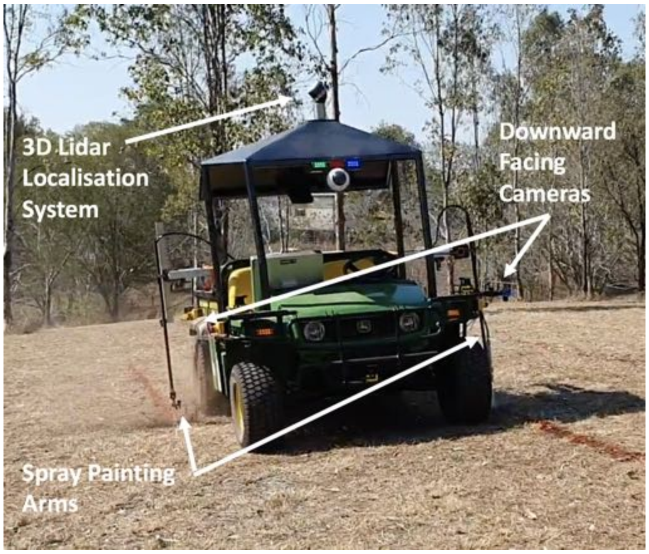


Figure 1: Overview of the autonomous platform used for field scanning and marking. The lidar localisation system is installed in preparation for future projects, and was not used in this paper.

the region, then retraces its path in reverse, before driving across to the next lane in what we will refer to as the 'dogleg' manoeuvre. This pattern is then repeated for each 'lane' that is scanned, as illustrated in Figure 3. This pattern avoids a more challenging turning manoeuvre at each end of the field, and allows the scans to take place on rectangles that have no space for turning at the far end.

Initially, as the vehicle travels forwards, it sprays the left side with white paint (as it has just scanned the ground and the area to the left of white is deemed as safe) and the right side with red paint (right side of red is unsafe). As it travels backwards, it simply follows the marked lines. It then moves forwards, onto the next lane to the right (the dogleg) where it oversprays the previously painted red line (now on the left) with white paint.

3.2 Choice of Camera and Sprayer Locations

Figure 4 shows the location of cameras and paint spray arms on the vehicle. These are positioned to optimise the ability to follow the painted lines in a stable manner.

The left camera is placed level with the front axle, so that wheel angle changes have a direct and immediate effect on the position of the sprayer, which is placed just behind the camera view. This avoids the overspray (white) interfering with the line detection, but also allows the camera to observe spray location in future upgrades. This is useful, for example, if winds are offsetting the sprayed paint slightly from where it is expected to land. The right camera is placed level with the back axle. This is the only point along the length of the car that is not offset laterally by steering the vehicle. It is therefore the most stable location to control to follow the red line, that was painted on the forward run.

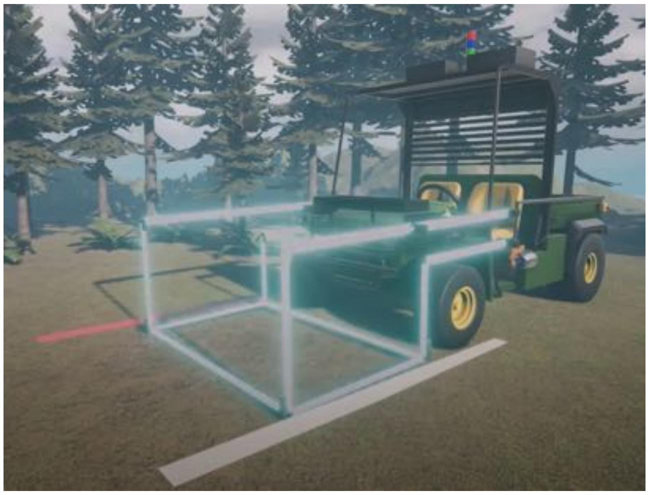


Figure 2: Illustration of the low-resolution magnetic sensor (highlighted in white) that can be mounted on the vehicle.

The right painter is less constrained in its position, but is best placed also next to the back right wheel. This is because the backwards line detection needs to have painted line all the way back to the start of each lane.

3.3 State Machine

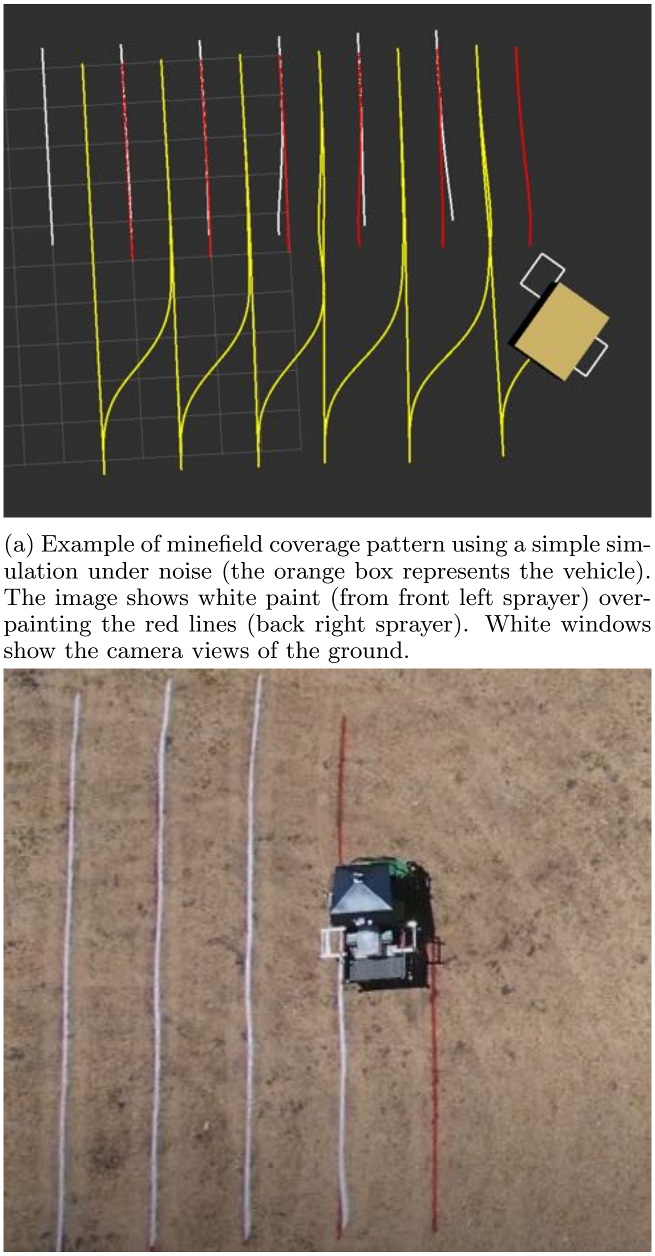
The control of the vehicle follows a hierarchical state machine, as shown in Figure 6. This means that each state can contain a finite state machine within it, to perform the more specific actions to achieve the goal of the parent state. In our case we have one base state (scan field), with the purpose of scanning the minefield according to the specified length and width of the field. This state acquires the pertinent sensory information and outputs the control messages to the motor controllers, transitioning out only when the expected width been scanned.

Within this are five child states that run in a cyclic pattern, representing the repeated behaviour to scan each lane, as visualised in Figure 3.

3.4 Line Detection

The task of the line detection module is to convert a downwards facing video stream of ground with a painted line on it, into a position and gradient of that line.

The task is modeled as a ridge-detection task for a particular known line colour (red or white). As with most outdoor image processing applications, the main difficulty is the variable lighting. It is expected that the



(a) Example of minefield coverage pattern using a simple simulation under noise (the orange box represents the vehicle). The image shows white paint (from front left sprayer) over-painting the red lines (back right sprayer). White windows show the camera views of the ground.

(b) Aerial view of the real vehicle performing the coverage illustrated in (a).

Figure 3: Illustration of the coverage and driving patterns.

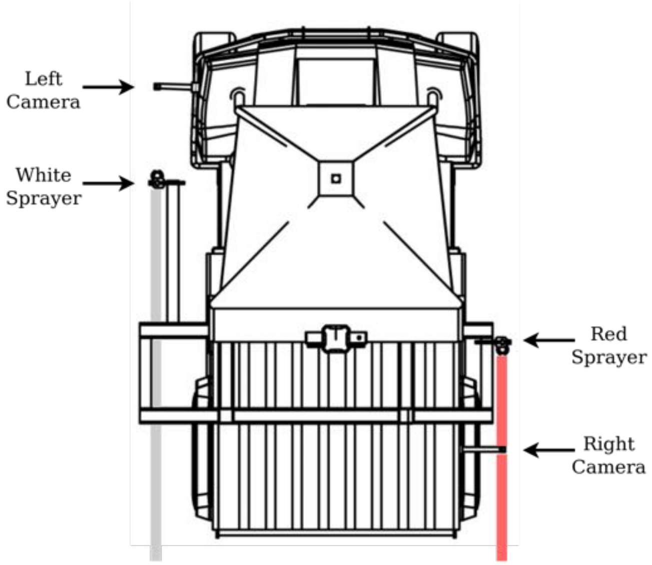


Figure 4: Plan view of the autonomous vehicle, highlighting the downward-facing camera and sprayer locations.

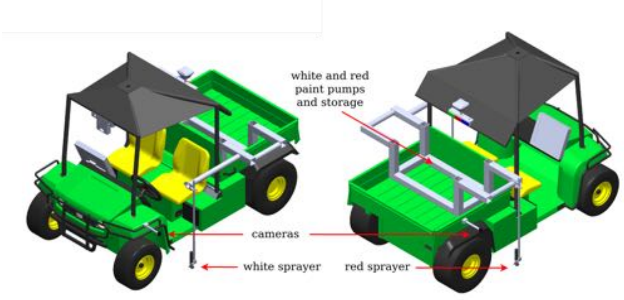


Figure 5: 3D illustration of the components shown in Figure 1.

vehicle may be running in any condition from low, overcast light, to strong sunlight. The camera pixel value of the painted line can vary greatly over this range, and indeed overlap the range of pixel values for the surrounding ground. There is not restriction on the background colour, and the system must handle anything from dusty earth to grey gravels and green grass colours.

The second challenge of lighting is shadows. For line detection it is particularly challenging because the straight-edged shadow of the vehicle will be apparent for some sun angles, which gives a strong straight edge that creates a false positive for many line detection methods.

To address these lighting challenges we apply an initial intensity function $i = f(r, g, b)$ that converts a three-dimensional RGB colour value into an intensity value i. This is specialised for the lighting conditions and paint colours that we are testing under. The function is:

$$i = [3/2 - 5|(r, g)/(r + g) - c|]_0^1 \tag{1}$$

where $[x]_0^1$ clamps x between 0 and 1, and c is the target colour: $(2/3, 1/3)$ for red lines and $(1/2, 1/2)$ for white lines.

This function ignores the blue component of the pixel colour entirely, as we found that it varies excessively with lighting changes. The remaining red and green components are divided by their sum, to make the function

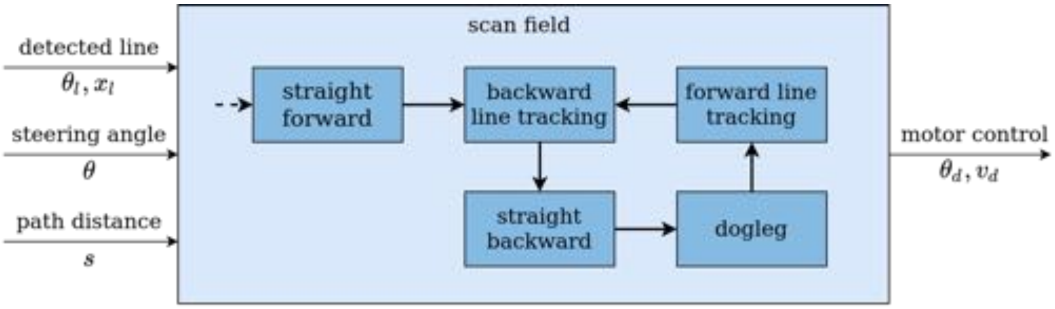


Figure 6: An overview of the hierarchical state machine for minefield scanning. ‘scan field’ is the parent state, with five child states shown as the darker blue boxes. External arrows indicate the input and output data. Bold arrows indicate transitions, and the dotted arrow indicates the initial child state.

independent of light level. The function therefore only discerns images by their red-green hue, and the proximity of this to the red-green hue of the target paint colour. The 3/2 value avoids the observed red-green hue needing to be a perfect match for the target hue, keeping the line intensity more consistent.

We now have an intensity image, with a signal of a high intensity 5-7 cm wide line, within a background of mixed intensity ‘noise’. The next step is to apply a Difference-of-Gaussians (DoG) filter to the signal, to convert the wide line into a more distinct peaked ridge. This DoG filter is a band-pass filter, which we configure to enhance the spatial frequency band corresponding to the 5-7 cm thick line. It therefore acts to convert the solid bar into a signal closer to a single ridge, while suppressing high frequency noise.

With this ridge we now apply the Hough Transform [Hough, 1962] to efficiently extract the candidate lines of best fit based on the intensity image. The results are shown in Figure 7.

3.5 Odometry

In the underlying ‘scan field’ state, we maintain a state estimation of the vehicle using wheel odometry. This integrates the measured wheel rotational velocities together with the steering angle and the Ackermann vehicle kinematics. This is odometry is used for estimating the initial pose for the ‘dogleg’ manoeuvre, and in estimating when to exit each state, based on distance travelled.

3.6 Forward Line Tracking

This task is to control the vehicle steering angle θ_d in order to keep the detected line in the centre of the camera image. This line has angle θ_l and the front wheels have mean average angle θ. These are all in the same coordinates, with zero being in line with the forward axis of the vehicle, and positive values driving the vehicle to the right.

This task is performed using a variation of the standard PID controller. It uses the distance from line centre to camera centre as the error to be minimised, and the steering angle as the control parameter. The D part of a PID controller is used to converge to a uniformly moving target. In our case, the rate of change of error is known from the current steering angle and the line angle, so we do not require a feedback D gain here, we just constrain the steer angle directly to equal the line angle: $\theta_d = \theta_l$.

Physically speaking, this is continuously setting the tyres to be parallel to the observed line. This controller will of course drift over time, so we include the P term, which adds an additional steering angle to track back towards the detected line position.

$$\theta_d = \theta_l - Pe \tag{2}$$

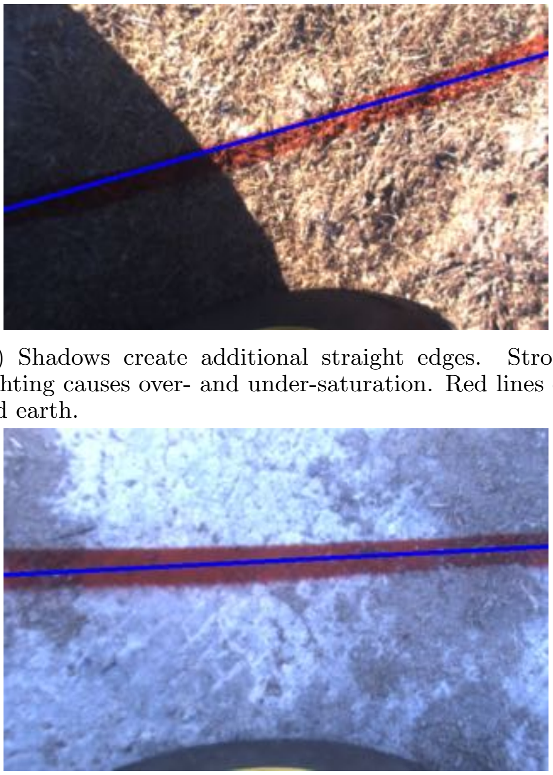
where e is the error, the difference between the mid-point of the detected line (x_l) and the centre of the camera image, which is inline with the sprayer location. Positive e means the vehicle needs to move further to the left.

As a kinematic controller, the vehicle should not oscillate. The P term applied to the steering angle, together with the parallel wheel angle term, should bring the vehicle into alignment in a manner that tends towards an exponential decay.

This of course does not happen in practice, as the commanded steering angle is not met, either immediately or precisely. There is a delay of up to half a second in the full feedback loop, from line detection, to commanded motor angle, to physical movement of the tyres. Nevertheless, we have found that when the P value is not overly large, and there are high gains on the low level steering controller, the oscillation is minimal.

3.7 Backward Line Tracking

With respect to path distance, forward line tracking is fundamentally a first-order problem: one can turn tyres to drive the front wheels at a chosen angle towards the



(a) Shadows create additional straight edges. Strong lighting causes over- and under-saturation. Red lines on red earth.

(b) Rocky ground, and at a different time of day, the blue content is very different.

Figure 7: Challenges that our line detection method had to overcome. Taken from field trials.

line. Backward tracking however, is a second-order problem: one cannot directly set the angle of back wheels towards the line, only its curvature. This makes it a more difficult tracking task in general.

We control the backwards line tracking using the following controller:

$$\theta_d = \tan^{-1}(Lk) \tag{3}$$

$$k = -\omega^2 eH - 2\omega\theta_l \tag{4}$$

where L is the vehicle length between axles, and H is the half-width of the camera view on the ground. This function is very similar to the critical damping function $\ddot{x} = -\omega^2 x - 2\omega\dot{x}$ which produces stable second-order tracking without oscillation. The difference is that it is operating on curvature (the second differential with respect to forwards distance), rather than the second differential with respect to time. The $\tan^{-1}(Lk)$ transforms the desired curvature k of the vehicle path into a mean front wheel angle, as expected from Ackermann steering.

As with the forwards line tracking, the controller is a function of the detected line lateral error e and the detected line angle θ_l, with negative feedback on the error component. However, unlike the forward tracking, the backwards tracking directs the steering angle in the opposite direction to the detected line angle. If the line angle tilts clockwise then the front wheels turn anticlockwise, because the vehicle chassis turns counter to the wheel angle in backwards motion.

For backward tracking, the feedback is less immediate, we therefore use a lower gain than forward tracking, specified as ω.

3.8 Dogleg

After tracking the white line on the left on driving forwards, and tracking the red line on the right in reverse, we then need to drive into the next lane to the right. This is the most difficult part as we move away from the painted line and rely on just wheel odometry until we find the new section of white line in the left camera view.

The ‘dogleg’ manoeuvre follows a spatial S-like curve of length d. We model it as a curvature that follows a sine wave $k(s) = a\sin(2\pi s/d)/L$ for some amplitude a at path distance s along the curve. At each time step during this motion, we find the nearest point on this curved path to the predicted pose of the vehicle. We then obtain the lateral offset in metres from the vehicle's left camera view centre on the ground to the curve (eH). We also calculate the angle between the path's tangent and the vehicle forward axis (θ_l), and the path's curvature at this point (k). Now we perform line tracking according to:

$$\theta_d = \tan^{-1}(kL) + \theta_l - Pe \tag{5}$$

This is similar to the forward line tracking, but includes the curvature term since the path has a well-defined curvature in this case.

3.9 Persistent Offsets

The above controllers work well in theory, but in practice there are misalignments and systematic errors that cause the vehicle to not converge to the line, but instead converge to some offset from the line. The cause of this may be internal (e.g. tyre pressures, camera misalignments) or external (e.g. sideslip when driving on a lateral slope), in both cases we can remove this offset by including the I (integral) term of the PID controller.

The I term requires some tuning. If it is too small then it will not adapt quickly enough when the cause of persistent errors changes, such as the slope changing. If it is too large then it introduces oscillation into the vehicle's motion. In each of the above controllers, we include this additional term to the motor steering angle: $[I\int e\ dt]^{0.5}_{-0.5}$

This I term has two issues of windup. Firstly, if a persistent error cannot be resolved then the I term can grow without bound, we fix this by providing limits on the I term, as shown above. Secondly, the integral term should not need to 'wind up' each run. We resolve this by calibrating the integral term due to internal effects on a flat field, and storing this term, for use as the initial value (the constant of integration). In fact we store a separate initial value for the forward and backward line tracking, as their dynamics are different.

3.10 Velocity Control

The high level control of velocity is simple. We keep the vehicle at a constant velocity of 0.6 m/s for the dogleg, forward tracking, and backward tracking, with a linear deceleration at the end based on a distance travelled estimation. That is, we prescribe a linear ramp down of velocity with proximity to the expected path distance for each manoeuvre. The distance travelled is an integration of the wheel velocity measurements from the start of each manoeuvre.

The initial acceleration phase is a consequence of the underlying vehicle velocity control gains, we do not specifically ramp them up.

3.11 Simulation

In order to test the robustness of our scanning method, we used a simple simulation of the Ackermann steering model under noise. This retrieves the motor control outputs (shown in Figure 6) and simulates the line detection, and new steering angle and path distance values that enter the 'scan field' base state. The steering and velocity of the vehicle are not immediate, they are modelled as an exponential decay towards the desired control θ_d and v_d. Noise is added to these two components and also to the location of the detected line.

Modifying these noise parameters helps in estimating how the scan pattern will develop under the more difficult conditions. In Figure 3 the initial gap between white and red lines after lane three demonstrates that the 'blind' dogleg manoeuvre is the part that is most sensitive under high noise levels.

4 Experiments

In this section we describe details of our hardware and software implementation followed by experimental results.

4.1 Hardware and Software Setup

Vehicle and Sensors

The vehicle used is an electric John Deere Gator TE automated by the CSIRO [Pfrunder *et al.*, 2017]. The vehicle has had an automation kit installed with SmartMotors to control the steering and brake actuation. The cameras used for line detection and tracking are Point Grey (Flir) Grasshopper 3 colour cameras.

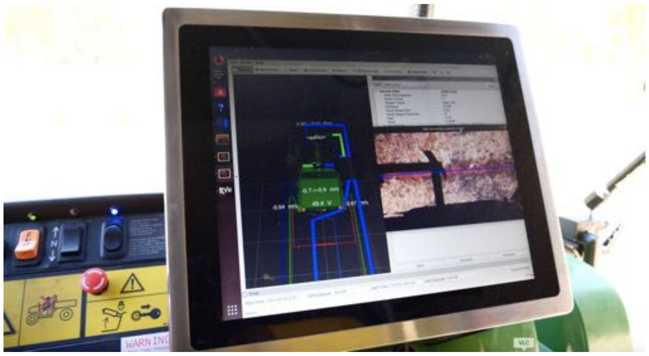


Figure 8: Onboard GUI to visualise line detection, vehicle trajectory and state.

Computing Hardware and Software

The Computing hardware used is a rugged embedded Neousys Nuvo-7006 PC with an Intel i7-8700 Processor. Distributed IO was performed by an Advantys STB system and the safety system runs on a PILZ PNOZmulti safety controller.

The system was running on Ubuntu 18.04 and the ROS (Robotic Operating System) middleware. The control system software and Advantys IO interface software is proprietary. Camera interface software is open-source through the ROS system.

Paint System

The painting system consists of two independent domestic airless wet paint spray systems (Wagner Control Pro 150). To activate and deactivate spraying, we installed a 24V high-pressure solenoid valve located before the nozzle. System power was supplied via a 1200W DC/AC inverter run from the test vehicles internal batteries. All components were mounted to an adjustable steel frame (shown in grey in Figure 5) secured to the back tray of the vehicle. Once mounted, the positioning of each spray nozzle was achieved by manual adjustment of clamping mechanisms on the steel frame. The spray systems solenoids were then connected and controlled via the vehicles distributed IO system.

4.2 Results

The first set of results measures the effectiveness of the line detection algorithm. We have manually observed the detection of red and white lines, over four runs, representing approximately 10 minutes of footage, and recorded the errors away from the correct line. There was high variability in the background, ranging from green grass to dirt and rocky terrain. The results are shown in Table 1. An illustration of the GUI with the real-time detection and vehicle state is shown in Figure 8.

Next, we present the full demonstration of the system: a scan of 10 rows of 12 metres in length, representing

Video type	Total time	Period with line present	Incorrect period
Red 1	167 s	141 s	0 s
Red 2	64 s	45 s	0 s
White 1	246 s	193 s	0 s
White 2	118 s	118 s	1.5 s

Table 1: Period of incorrect line detection (not overlapping the painted line). This represents 0.3% of the time when lines were in view.

a minefield of approximately 231 m^2 in area. Operationally, the fact that we lay down paint makes it very difficult to perform multiple runs (as once the field is painted it cannot be used again), so the tests leading up to this result were performed in different fields. We present this single long run, rather than multiple short runs because it is important to observe how the vehicle behaves when scanning a large number of rows, and to what extent the vehicle path drifts from row to row.

For this single large scan, we have taken an aerial image (Figure 10) and are able to obtain error metrics by manual pixel counting of any remaining red lines that were not fully covered by the white painted line. For each length where red line is still visible, we estimate the mean pixel width of that sliver of uncovered paint, and its length along the line. Adding these up and multiplying by the image's pixel scale (1.04 cm) provides our measures of error for this experiment. These error statistics are displayed in Table 2. While this shows some perceptible alignment errors, they are not large enough to affect the success of the operations, with the vehicle covering and marking the field according to the requirements. Figure 9 shows that the errors do not impact the performance of the marking system.

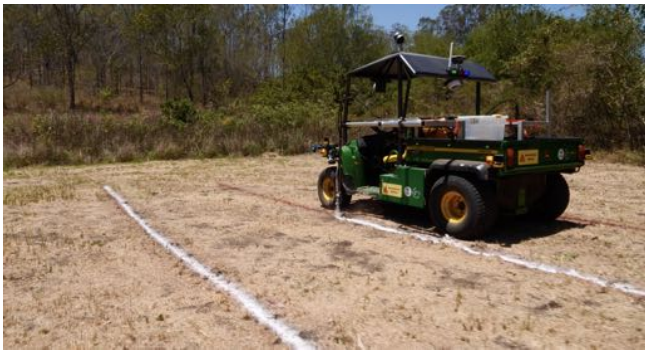


Figure 9: Illustration of the 'live' spraying of white paint over red paint. The small precision errors shown in Table 2 do not affect the practical performance of the marking system.

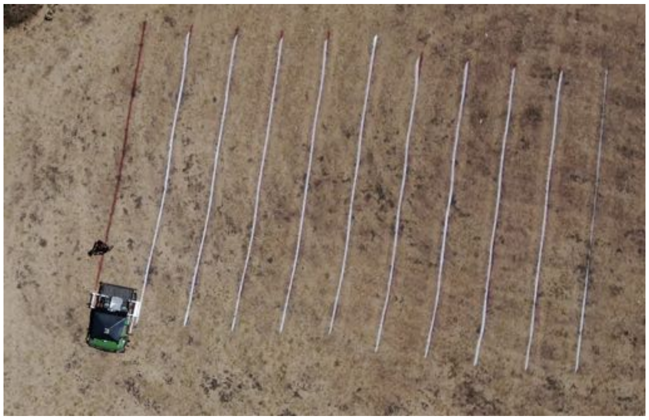


Figure 10: The result of scanning 10 rows of an area representing a minefield, by line following each painted lane in turn.

Measure	Value
Number of rows	10
Total line length	120 m
Mean line length per row	12.0 m
Area scanned	231 m^2
% of line with any visible offset	25.3%
Maximum error	10.0 cm
Mean Absolute Error	0.88 cm
Root Mean Square Error	1.82 cm
Mean change in bearing per row	0.532^o

Table 2: Estimated errors by manually measuring thickness of remaining visible red lines in Figure 10.

4.3 Discussion

Looking at the resulting scan in Figure 10, the largest errors occur at the top. This is expected, as this is where the dogleg manouevre ends and the tracking of the red line resumes. This dogleg relies entirely on wheel odometry, so is expected to have some variation.

Overall, the lines remain almost straight, which is important. We believe one reason for this stability is the positioning of the sprayers and cameras. The vehicle tracks the left-side red line at the front axle position, while spraying the right-side red line at the back axle position (see Figure 4). Due the front-wheel (Ackermann) steering, the painted red line acts physically as a low-pass filter on the left-side line tracking. This keeps the lines smooth, and prevents buildup of line deviations from row to row.

A video illustrating the vehicle operating and the overall concept is available at `https://youtu.be/JfEoLkIYQ84`

5 Conclusions

We have illustrated that a minefield scanning vehicle is capable of precise coverage using only wheel odometry and painted line tracking. This is an important baseline result, because additional forms of localisation (SLAM, GPS, INS) have failure cases, such as in featureless environments or canyon areas. We therefore need to be able to rely on minimal sensing options for whenever more advanced localisation is not suitable.

Minefield clearance requires a visual marking of cleared areas and the proposed system with spray painting from the vehicle is one of the clearest and most direct ways of achieving this. It follows that the vehicle must accurately scan within the lines it has itself sprayed, and therefore localisation by line tracking is almost a direct consequence of the requirements of minefield clearance. We have demonstrated that this system does work with sufficient accuracy in real-world environments.

In future we intend to tackle the more challenging situations of obstacles in the field, more hilly ground, and non-rectangular clearance regions. This will require the fusion of the baseline localisation with SLAM and GPS systems, and will require planning algorithms to determine the coverage pattern.

References

[Baudoin and others, 2008] Yvan Baudoin et al. Mobile robotic systems facing the humanitarian demining problem state of the art (sota) december 2007 itep 3.1. 4 task. *The 7th IARP International WS HUDEM*, pages 28–30, 2008.

[De Cubber *et al.*, 2014] Geert De Cubber, Haris Balta, and Claude Lietart. Teodor: A semi-autonomous search and rescue and demining robot. In *Applied Mechanics and Materials*, volume 658, pages 599–605. Trans Tech Publ, 2014.

[Doroftei *et al.*, 2014] Ioan Doroftei, Romain Malherbe, Gaetan Bred, Yvan Baudoin, and Ionel Conduraru. Remote controlled wheeled mobile robot for humanitarian demining purposes. In *Applied Mechanics and Materials*, volume 658, pages 618–625. Trans Tech Publ, 2014.

[Estremera *et al.*, 2010] Joaquín Estremera, Jose A Cobano, and P Gonzalez De Santos. Continuous free-crab gaits for hexapod robots on a natural terrain with forbidden zones: An application to humanitarian demining. *Robotics and Autonomous Systems*, 58(5):700–711, 2010.

[Fukushima *et al.*, 2005] Edwardo F Fukushima, Paulo Debenest, Yuki Tojo, Kensuke Takita, Marc Freese, Helmuth Radrich, and Shigeo Hirose. Teleoperated buggy vehicle and weight balanced arm for mechanization of mine detection and clearance tasks. *Proc. of the HUDEM*, pages 58–63, 2005.

[Gonzalez de Santos* *et al.*, 2005] Pablo Gonzalez de Santos*, Elena Garcia, Joaquin Estremera, and Manuel A Armada. Dylema: Using walking robots for landmine detection and location. *International journal of systems science*, 36(9):545–558, 2005.

[Hemapala *et al.*, 2009] Manjula Hemapala, Vittorio Belotti, Rinaldo Michelini, and Roberto Razzoli. Humanitarian demining: path planning and remote robotic sweeping. *Industrial Robot: An International Journal*, 2009.

[Hough, 1962] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[Huang and Nonami, 2003] Qing-Jiu Huang and Kenzo Nonami. Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control. *Mechatronics*, 13(8-9):773–790, 2003.

[Ian, 2017] International campaign to ban landmines, landmine monitor year (icbl-cmc: 2017). 2017.

[Marques *et al.*, 2016] L Marques, AT De Almeida, Manuel Armada, Roemi Fernández, Héctor Montes Franceschi, P González, and Y Baudoin. State of the art review on mobile robots and manipulators for humanitarian demining. 2016.

[Pfrunder *et al.*, 2017] Andreas Pfrunder, Paulo VK Borges, Adrian R Romero, Gavin Catt, and Alberto Elfes. Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3d lidar. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2601–2608. IEEE, 2017.

[Portugal *et al.*, 2014] David Portugal, Lino Marques, and Manuel Armada. Deploying field robots for humanitarian demining: challenges, requirements and research trends. In *Mobile Service Robotics*, pages 649–656. World Scientific, 2014.

[Santana *et al.*, 2007] Pedro F Santana, Jose Barata, and Luis Correia. Sustainable robots for humanitarian demining. *International Journal of Advanced Robotic Systems*, 4(2):23, 2007.

[Suh *et al.*, 2014] SeungBeum Suh, JunHo Choi, ChangHyun Cho, YeonSub Jin, Seung-Yeup Hyun, and Sungchul Kang. Mine detecting robot system. In *Field and Service Robotics*, pages 97–109. Springer, 2014.

[Trevelyan, 1997] James Trevelyan. Robots and landmines. *Industrial Robot: An International Journal*, 1997.

[Won *et al.*, 2001] IJ Won, Dean A Keiswetter, and Thomas H Bell. Electromagnetic induction spectroscopy for clearing landmines. *IEEE Transactions on Geoscience and Remote Sensing*, 39(4):703–709, 2001.