User interface and coverage planner for agricultural robotics

Daniel Richards[1], Tim Patten[2], Robert Fitch[2], David Ball[1], Salah Sukkarieh[2]
[1]School of Electrical Engineering and Computer Science
Queensland University of Technology (QUT), Australia
[2]Australian Centre for Field Robotics
The University of Sydney, Australia
d7.richards@connect.qut.edu.au

Abstract

Farmers are under growing pressure to increase production, a challenge that robotics has the potential to address. A possible solution is to replace large farm machinery with numerous smaller robots. However, with a large number of robots it will become increasingly time consuming for the farmer to monitor and control them all, hence the need for an effective user interface and automatic multi-robot coordination. This paper describes the design of a user interface and coverage planner suitable for controlling multiple robots for typical coverage style farm operations. The cross-platform user interface allows the farmer to specify their farm including fields, roads and docking stations. The coverage planner splits the workload between the robots and plans periodic docking. The results for the different multi-robot coverage strategies demonstrate the advantage of the robots sequentially moving between fields rather than freely moving between them. The multi-robot system has been used for a coverage task on a real farm for controlling two real robots and four simulated robots operating for two days.

1 Introduction

In modern agriculture farmers face a wide variety of problems such as herbicide resistant weeds [Heap, 2014], nutrient management, soil compaction [Hamza and Anderson, 2005] and an ageing workforce. Advances in technology, such as robotics, can potentially address these issues by reducing individual workload, monitoring the environment and reducing soil compaction.

Researchers such as [Blackmore *et al.*, 2001] and [Pedersen *et al.*, 2006] have proposed replacing large heavy farm machinery with multiple small and lightweight autonomous robots. However, for robotics to become widely accepted within the agricultural industry the robots must be easy to control and monitor. Furthermore, there are issues to consider about the most effective way to deploy the robot system on a farm.

A multi-robot system operating in the agricultural environment will often have vehicles travelling large distances, working in remote locations and out of line of sight. In this situation it is important for robots to be self-sufficient and able to refuel themselves at docking stations. Additionally, the user must be able to determine the state and current plans of any vehicle within the system quickly and accurately. While this information could be displayed through a text based interface; this will not be appropriate for an ageing workforce.

This paper describes the design of a multi-robot system to control agricultural robots for coverage tasks. The system consists of a graphical user interface (GUI), coverage planning system, and an emergency stop system appropriate for the farm environment. The user interface is able to visualize the state of multiple robots and allows configuration of the farm such as field boundaries, roads and docking stations via a map image. Additionally, coverage plans for each robot are dynamically created based on the operating environment defined by the user. Moreover, the user interface is able to run on a variety of platforms including Windows, Android, Apple and Linux. The user interface and coverage planner were tested in a two day field trial in which four simulated and two real vehicles performed a coverage task in a 59-ha field.

Further we investigate the feasibility of the system compared to a single large human-operated tractor in terms of area covered over time. Robots move slowly and have a narrow working area, but can operate 24 hours a day. The paper presents multi-robot coverage algorithms adapted to broad-acre agriculture and report whole-farm simulation. The algorithms consider the robots' docking to replenish consumable spray resources. The paper also compares two methods of distributing the robots amongst the fields. The first is to move the team of robots and a single docking station as a single unit between fields. The second is to allow the robots to work on any field on the farm, with numerous docks spread throughout the farm.

In previous work we demonstrated an autonomous robot performing a coverage task using inexpensive sensors for vision-based obstacle detection and localisation by [Ball *et al.*, 2013]. However, this system was configured and monitored via terminal commands and text files and didn’t have docking capability. This work replaces the manual configuration with a graphical user interface and includes results around docking.

The remainder of this paper is organised as follows. Section 2 reviews the literature in multi-robot coverage, user interfaces, and web technology. Section 3 discusses the design approach to creating a graphical user interface for the farmer to control the robotic platform and the algorithm for coverage. Section 4 presents the results of both functional and experimental testing of the system. This paper is then concluded in section 5.

2 Literature

This section reviews some of the current robot and agricultural interfaces on the market and the technologies which can be used to create such interfaces. Furthermore approaches to multi-robot coverage mapping are discussed.

2.1 Multi-robot coverage

Single robot coverage planning has a rich history in the robotics community. The variety of approaches that have been proposed, including cell decomposition methods, grid-based methods, and graph-based methods, were recently surveyed by Galceran and Carreras [Galceran and Carreras, 2013]. An earlier survey was presented by Choset [Choset, 2005]. Coverage planning is related to the Travelling Salesman Problem (TSP) and is NP-hard for both the single-robot [Galceran and Carreras, 2013] and multi-robot [Rekleitis *et al.*, 2008] variants.

The approach to multi-robot coverage that we apply in this paper is an exact cell decomposition method based on the classical *boustrophedon decomposition* [Choset, 2001]. Recent work by Xu, Viriyasuthee, and Rekleitis [Xu *et al.*, 2014] shows that the order of cell traversal can be optimised using a Reeb graph representation, where edges represent cells and vertices represent critical points. However, this algorithm splits cells by introducing mid-row turns and cannot be applied to our case, where turns are restricted to row end points.

Recent work on coverage planning in the context of agricultural robots focuses on the problem of choosing an optimal track orientation [Hameed, 2013; Jin and Tang, 2011; Xu *et al.*, 2014]. In this paper, the track orientation is determined in advance by existing planting patterns and therefore basic results in coverage may be applied. Our focus instead is on the whole-system aspects, including navigation and perception, that allow these results to be applied in practice.

Refill planning (servicing) is an extension to coverage planning that considers the case where a robot must temporarily suspend coverage execution and travel to a docking station to replenish its physical resources. In spraying applications, robots expend resources such as herbicide and energy during coverage; coverage planning must therefore consider physical constraints such as limited volume of spray tanks and limited on-board energy supply. Refill planning allows robots to satisfy such constraints and is critical for long-duration operation. However, there has been relatively little exploration to date of efficient (polynomial-time) algorithms for this problems and their performance in practice. A taxonomy of problem variants is proposed in [Bochtis and Sørensen, 2009] and [Bochtis and Sørensen, 2010] along with a mapping to variants of the vehicle routing problem (VRP). As the authors note, the VRP and its variants are NP-hard in general.

A simpler, polynomial-time approach to refill planning is proposed by Oksanen and Visala [Oksanen and Visala, 2009], where refill trips are chosen greedily for a single robot and single field based on capacity estimation.

2.2 User interfaces

Since the 1940s humans and robots have been working together to achieve complex tasks. Research in the area of multi agent robotics systems has led to the need to be able to simultaneously control multiple platforms from within a single user interface [Laengle and T.Hoeniger, 1997].

Typically in the agricultural environment; farming equipment includes a GUI to manage and monitor the operation of the vehicle. Many parameters related to subsystems such as transmission, hydraulics, lighting, hitch, power take off and radios all must be shown to the user. For controlled traffic farming, where the farm vehicle is automatically steered along pre-defined paths, the user can see the vehicle’s path in real world coordinates. However, the user interface still focus on configuring a single farm machine.

When developing user interfaces there are two approaches: web-based and a native application. The web-based approach is the development of a website in languages such as HTML, JavaScript, CSS and PHP and take advantage web browsers to provide support for multiple devices. This approach requires a webserver that the user can connect to via the internet or a local network. Using technologies such as NodeJs and RosLibJS on the server side application can allow for an interface between the robot middleware such as ROS. This approach requires only a single set of source code to provide functionality to multiple platforms but is susceptible to changes in browser support which can make a package obsolete.

An example of a web-based GUI for a robotic platform is Mavelous, a basic user interface for sending commands to a UAV using the MAVlink protocol.

Alternatively, native applications are developed for a specific device and can be optimized to run more efficiently. Native applications can also access device specific hardware. Typically these applications are precompiled and installed on the target device [Charland and Leroux, 2011]. Each platform requires a software development kit and framework [Heitkotter, Hanschke, and Majchrzak, 2012]. Native applications can also employ a variety of networking techniques to communicate to any robot middleware.

In the field of unmanned aerial vehicles there are many native GUI’s. Two commonly used are Qgroundcontrol and Mission Planner which are open source control stations which support the use of 2/3D maps provided by a chosen

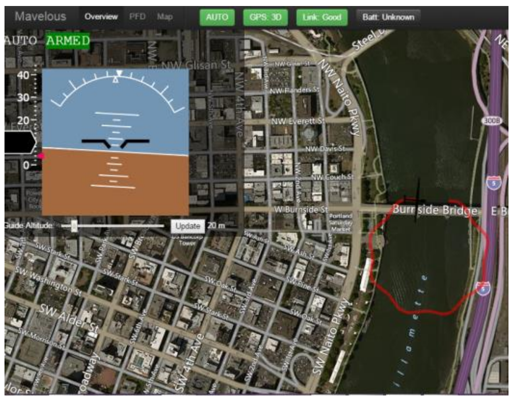


Figure 1. Mavelous web based user interface for controlling a UAV.

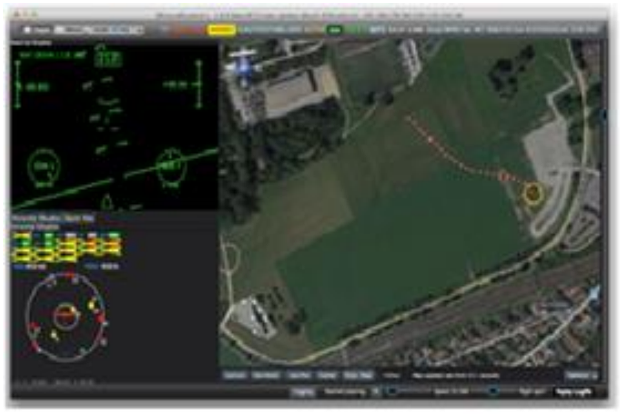


Figure 3. Qgroundcontrol user interface for UAVs.

mapping service. The user is able to define paths they wish there vehicle to travel by clicking locations on the map and setting up waypoints. Both interfaces support up to 255 simultaneously operating vehicles but require the user to define a specific plan for each. While both GUI's can be used with many ground based vehicles they have been optimised for use with Aerial vehicles, which is evident when looking at the main control page where devices for monitoring flight parameters can be seen.

The native approach to multiplatform development is often costly and requires multiple developers with different programming experience. However, cross platform IDEs have been developed that allow the user to compile for many different platforms from a single set of source code. Qt creator is a cross platform IDE that allows the user to create graphical user interfaces using C++ , JavaScript or QML and can compile across a number of target platforms.

Unity3d is a cross platform game engine with unparalleled cross platform support currently supporting over 20 different platforms. Unity3D makes use of the C# language and its own development environment to create software. Furthermore, as a game engine Unity3D has a variety of inbuilt user interface and 3D rendering capabilities alongside its own networking protocol.

3 Design

This section presents an overview of the system including the technology chosen for development of the user interface and the coverage planner. Furthermore an overall system design and details about the implementation of the user interface, server technologies, safety radio and coverage planner systems will be presented.

3.1 System design

The overall system, shown in Figure 2, consists of 5 main elements the farmer, user interface, server, coverage planner and robot. The user interface must be able to visualize the state of the each vehicle in the field and provide a simple method for setting up the operating parameters required by the coverage planner. Furthermore, the GUI should be cross platform able to run on Linux. Windows, Android and IOS to ensure the user can access the system at any time.

Criteria such as supported platforms, hardware interaction, ease of development, GUI design and licensing fees were applied to Unity3d, Qt creator, web based solutions and individualized native solutions. Comparison

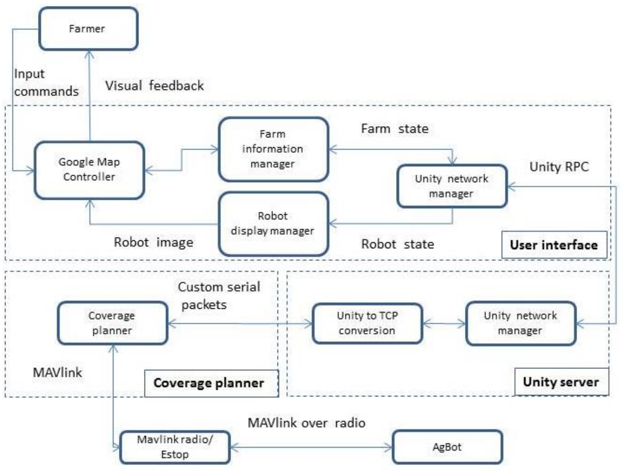


Figure 2. Overview of the multi-robot system. When a farmer interacts with the user interface messages are forwarded to the coverage planner and robots are updated.

of these approaches showed that Unity3d would be the ideal approach largely due to its ability to its cross platform support, inbuilt networking and rendering capabilities for GUI.

The farmer configures the farm environment which the robot operates. The state of the farm is then sent via Unity3D's network manager to a central pc. The coverage planner then uses the configured farm to generate plans for each robot. The plans are then issued to each robot via MAVlink messages. Modified RFD900 radios provide the means to transmit message between the robots and the coverage planner. The radios also provide an emergency stop system. Information about the state of each robot is transmitted back through the system to the robot display manager where an image is overlayed in Google maps.

3.2 User interface

The user interface was designed to convey important information to user in an easily understandable way. Each robot is displayed as a sprite and has two bars that represents the battery level and spray levels. The bars shrink and expand as the levels change. As shown in Figure 4 the sprite changes between Error, Stop and Active depending on the state of the robot.

The plan for each robot is visualized by drawing links between all future and previous way points for each robot. The path is drawn in a different colour for each robot that is operating within the same field and draws the previously covered path in red as shown in Figure 5.

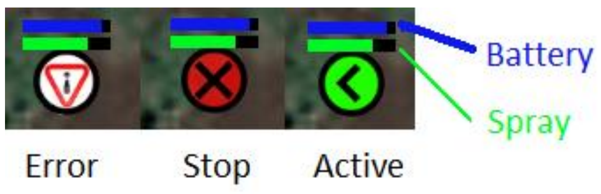


Figure 4. this figure show the three different states the robot can be in along with the battery and spray levels of each robot. The arrow shows the robot's current heading.

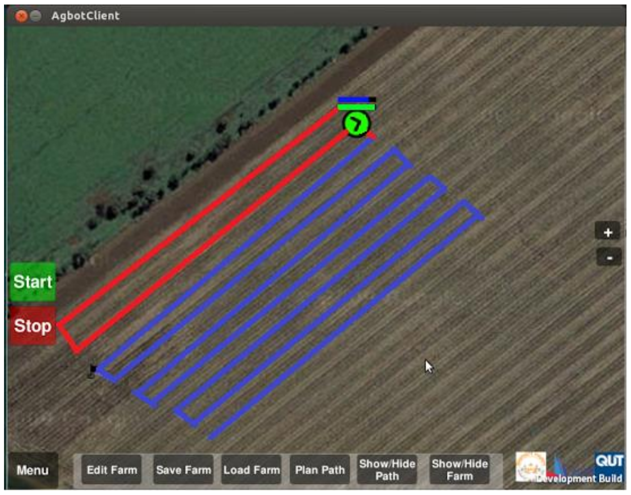


Figure 5. Screenshot of the user interface displaying one active robot, the already travelled path in red and the desired path in blue.

The user interface provides the user with an easy method to input a variety of different parameters that affect the way the robots operate on the farmland. There are three main features that are used for defining a robot farm: fields, roads and refuelling stations.

Fields define the boundaries of the area of which the robot must operate. This can be defined by 3 or more points placed on the screen by clicking the locations on the map image. Each point of the field is given an id number with the field and a field name so that multiple fields can exist within a single saved farm.

Each of these points must be accurately placed. As a point is placed on the screen it can be dragged into a new position by clicking and dragging. Alternatively, by selecting a point which has already been placed an additional information panel is opened and the latitude and longitude can be set manually. The first two points placed within a field define the angle of the crop rows on the field and thus the direction in which the robot will travel.

Roads define paths for the robots to travel between different fields or to their refuelling stations. Roads are defined by two or more points and represented by a red line between the points Figure 6

Refuelling points define the location which robots can travel to in order to recharge their batteries or refill their spray containers. Each refuelling location is defined by a latitude and longitude and a facing direction measured from due north. The combination of these three input types allows for all the required information to be passed into the coverage planner.

3.3 Server Communication

In the user interface-server side of the system the networking is primarily handled by Unity3D's inbuilt networking functionality. This networking implementation utilizes a TCP connection to allow two separate Unity3D applications to communicate.

The Unity3D networking operates via remote procedure calls (RPC) which operate as a standard function call which can be triggered from an external device. Each RPC call contains a target user allowing for multiple clients to be connected.

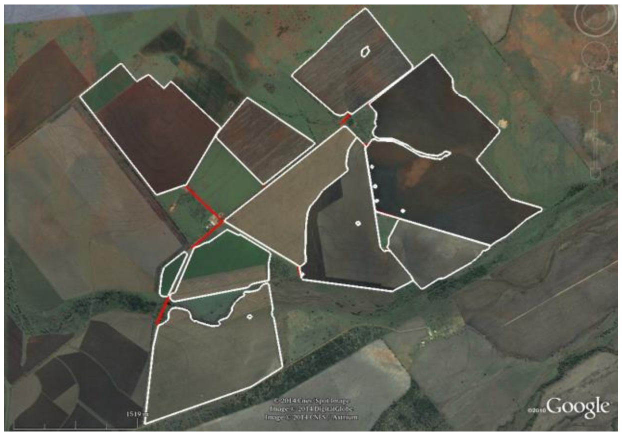


Figure 6. A representative farm layout used in whole farm experiments. White lines indicate field boundaries. Red lines indicate segments of the inter-field road network.

3.4 Coverage planner

The coverage planner accepts messages from the GUI which define; a field or fields represented by a geo-referenced bounding polygon with known row spacing and orientation and a geo-referenced path which represents the road network interconnecting the fields. The start location of m robots is also passed into the system via radio messages received from each vehicle. Additionaly, it is assumed any known obstacles within a field are represented as bounding polygons.

Using these parameters the task of weed management through controlled herbicide delivery is algorithmically an instance of the *coverage* problem [Choset, 2001]. This section describes the planning subsystem for multi-robot coverage of large fields. The coverage problem is one of finding an optimal path so that the robot will eventually cover all points throughout the defined area. Typically the coverage problem is related to the Travelling Salesman Problem (TSP) and is NP-hard [Galceran and Carreras, 2013]. Fortunately, in our application, vehicle motion is constrained to follow pre exisiting rows defined by the user. This removes the need to choose the optimal row orientation as in [Oksanen and Visala, 2009], and allows the adaption of exisiting coverage algortihims.

This section further describes the planning subsystem for multi-robot coverage of large fields.

3.4.1 Single Field

We apply the boustrophedon decomposition, where the coverage area is exactly partitioned according to a back-and-forth (lawnmower) pattern. This algorithm is described fully in [Choset, 2001] and summarised here for convenience. First, an exact cell decomposition is computed using a standard sweep line approach. An adjacency graph is then computed over the resulting partitioning. The order of cell coverage is computed using depth-first traversal of the adjacency graph. Finally, a coverage path is constructed from the resulting cell ordering.

In our case, the row orrienation is define by the user. Therefore, the orientation of the sweep line is determined and moves perpendicular to the row direction. An initial coverage path is generated starting from the cell that contains the robots' starting location. The path is represented as a sequence of row end points and road network waypoints. Path construction proceeds according to three cases. (1) Within a cell, the path consists of straight line segments (rows) connected in a back and forth manner with short inter-row segments. (2) Adjacent cells are also connected with short segments between adjacent rows. (3) Non-adjacent cells are connected by a path through the road network, including cell boundaries. These connecting paths are chosen to minimise the Euclidean distance between the non-adjacent cells. After the path is constructed, it is divided into m equal length sub-paths (one for each robot). The division made are such that the expected travel distances and times are equal for all robots. Diagrammatic examples of the type of cells generated by the boustrophedon decomposition can be found in [Choset, 2005].

3.4.2 Multiple Fields

In the multiple field case, each field is treated independently having its boustrophedon decomposition computed as outlined for the single field case. The result is an exact partitioning (represented as a set of cells) and adjacency graph for each field. These independent adjacency graphs are then combined to form a single adjacency graph that represents the entire farm.

To combine adjacency graphs, we first preprocess each independent adjacency graph by labelling each node that corresponds to a cell (in the exact partitioning) that lies on a field boundary. More precisely, a cell lies on a field boundary if one of its edges is coincident with a field edge. (Recall that a field is represented as a bounding polygon.) Then, for each pair of fields connected directly through the road network, we add edges to the corresponding adjacency graphs that connect labelled (boundary cell) nodes: each boundary cell in the first field of the pair is connected to every boundary cell in the second.

After the adjacency graphs are updated, the cell coverage order and path construction proceed as in the single-field case. This process results in a coverage path for each robot such that the entire set of fields is covered.

In practice, our coverage method provides flexibility in how multiple robots are applied to a given agriculture scenario. When a set of fields is input to the algorithm, the robots will naturally disperse and cover multiple fields simultaneously. This would be useful if the individual field size is small enough to be covered easily by one or several robots. At the other extreme, fields may be input to the algorithm sequentially to force all robots to operate together on each one. This is useful for very large fields where it is desirable to cover one fully before moving to the next.

3.5 Emergency stop radios

RFD900 radios are used to provide communication between robotic vehicles and the coverage planner in the form of MAVlink messages. In order to improve safety of the system, radios on each end where modified to also pass a basic heart beat message and to include an emergency stop button which immediately halts robot operation.

4 Results

Two key experiments where initially performed in simulation to test the coverage planner in a multi-field multi-robot situation. First the sequential approach in which each field is considered independently is tested. Robots complete the coverage of one field before moving to the next in this approach. This is then compared to the integrated approach in which robots are allowed to cover multiple fields simultaneously. Additionally, to test the capabilities of the GUI and coverage planner a two day field trial was completed using two real and four simulated robots operated over a 59-ha field.

4.1 Sequential coverage planning

As we only have a limited number of real robots, we validated the multi-robot coverage component of the system in simulation. First, we examined the behaviour of the system operating on a representative farm comprising multiple fields. Then, we compare two approaches to covering the farm where the first considers each field independently and the second considers all fields simultaneously. We also discuss implications of our results with respect to the value of multiple refill stations. We consider spray tank refilling only since tank capacity is the dominant resource constraint in the system, i.e. the spray tank has less capacity than the energy storage system and must be replenished more frequently.

The farm layout in the simulations is based on a successful cropping enterprise in Queensland, Australia. The farm area is partitioned into ten fields connected by a road network. Field boundaries and road segments were hand-labelled based on a geo-referenced aerial image. The total area of the fields is 1145 ha. The waypoint sequences output by the algorithm are passed to robot controllers that simulate the behaviour of the robots following the waypoints and log relevant data for analysis.

The boom sprayer is supplied by a tank with 200L volume. We assume an average spray rate of 40L/ha based on a typical application rate and dilution ratio. With 5m coverage width this results in 2km linear travel distance per hectare, or 10km travel distance to fully empty the spray tank. Thus a single robot can travel 10km (while spraying) before needing to refill. Robots travel at 5km/h constant velocity both while performing coverage and while traversing the road network. Travel between fields and refill stations is restricted to the road network and field boundaries; in order to refill, a robot must follow a row to its endpoint (which always falls on a field boundary) and then follow any combination of field boundary segments and road network segments to reach a refill station.

Our first experiment examines the behaviour of a robot team in a realistic farm scenario. We consider teams varying in size from one to twenty robots and compare their coverage performance. This experiment is important in that it informs the engineering tradeoffs that must be made in a practical application of the system, such as the number of robots employed versus time required to complete a spraying operation.

In this experiment, each field is input to the coverage algorithm independently and sequentially; the team works together to complete coverage of a single field before moving on to the next. The field ordering was manually determined. This use case represents how a farmer would use the system in practice if only one refill station were available, and the refill station must be manually towed to move it close to where the robots are working. We refer to this case as *sequential* coverage.

The simulated refill station is placed on a field boundary, coincident with a row endpoint. The refill station is placed to approximate the midpoint of one edge of the field boundary, with an equal number of rows on either side of the chosen position. Positions are determined a priori.

4.1.1 Sequential Results

Summary statistics are shown in Table 1 for teams of one, five, ten, and twenty robots. The mean area covered, number of refills, and distance travelled per robot vary inversely with team size as expected. The total time to cover the entire farm decreases from 520.3 hours (21.7 days) in the one-robot case to 65.0 hours (2.7 days) for 10 robots, and 36.6 hours (1.5 days) for 20 robots. Refilling at a docking station is modelled as an instantaneous operation in order to isolate the travel time component, which is the focus of this study.

A large tractor with a 36-m spray boom operating at 20 kph on average typically covers 50 hectares per hour, or 23 hours of working time (2-3 standard working days for a manual operator) to cover the 1145 ha farm. Therefore the ten-robot team best approximates the working rate of current standard operation.

Example coverage in the ten-robot case is shown in Figure 8. Screenshots show progress at four time points. The area covered by each robot is indicated by colour. The resulting coverage pattern shows the behaviour of the algorithm in dividing the field into partitions of roughly equal area.

The travel distance attributed to refilling is roughly 200 km in total, or 10% of total travel distance. This result shows that in a realistic application, considerable time and energy would be spent while moving to and from the refill station. Reducing this non-work component of the system's time performance could be approached by adding further refill stations or optimising their locations. These considerations motivate future work in this area.

Table 1. Summary statistics for simulated coverage of ten fields (shown in Figure 8) with varying team size. The total area covered in all cases is 1145.0 ha. Means shown are per robot, with standard deviations shown in parentheses. The final column summarises travel distance attributed to refilling (i.e., travel between row endpoints and the refill station). Speedup is the relative time-performance improvement due to multiple robots operating in parallel; the overhead (difference between the speedup value and the number of robots) is due to additional time incurred in travelling to refill stations and between fields.

Number of robots	**1**	**5**	**10**	**20**
Mean area covered (ha)	1145	229.0 (8.4)	114.5 (5.9)	57.3 (3.3)
Mean distance travelled (km)	2290	458.0 (16.9)	229.0 (11.7)	114.5 (6.5)
Total time to completion (h)	520.3	118	65	36.6
Speedup (x)	1	4.4	8	14.2
Mean number of refills	230	45.9 (1.8)	22.8 (1.6)	11.1 (0.9)
Mean distance travelled during refilling (km)	201.2	41.5 (15.9)	20.6 (9.4)	10.2 (5.2)

4.1.2 Comparison of Sequential and Integrated Coverage

The second experiment compares sequential coverage with *integrated* coverage, where the full set of fields is input to the algorithm. Intuitively, we expect integrated coverage to be more time-efficient than sequential coverage since integrated coverage does not force each robot to visit each field and hence incur the cost of the associated inter-field travel time. However, the integrated case would require the availability of multiple refill stations, since robots may operate in many fields simultaneously. Furthermore, it is not expected that either solition will be optimal.The purpose of this experiment is to compare the efficiency of integrated and sequential coverage in order to inform the need for multiple refill stations in a practical application of the system.

In this experiment we assume a team size of ten robots, which is a adequate for this farm based on the results of the first experiment. We ignore refilling so that we can isolate the difference in travel time due to inter-field travel.

An example of the coverage pattern at four time points is shown in Figure 7. The pattern again shows that each

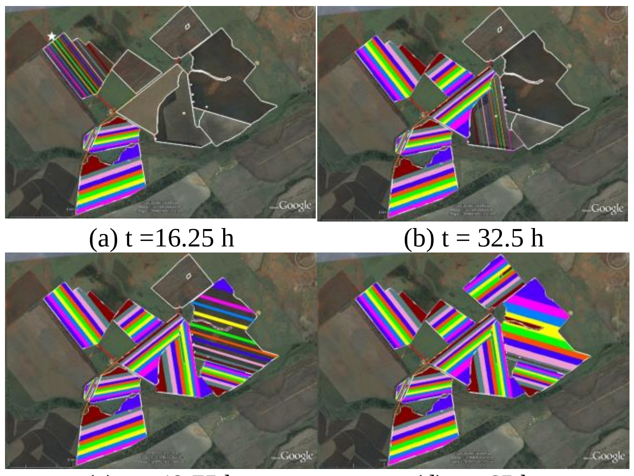

(a) t =16.25 h (b) t = 32.5 h
(c) t = 48.75 h (d) t = 65 h

Figure 8: Sequential coverage with ten robots at four time points. The cumulative area covered by each robot is indicated by coloured shading. The elapsed times are shown next to the figure identifier. The refill station location for one field is indicated by the white star. Other refill stations are chosen similarly.

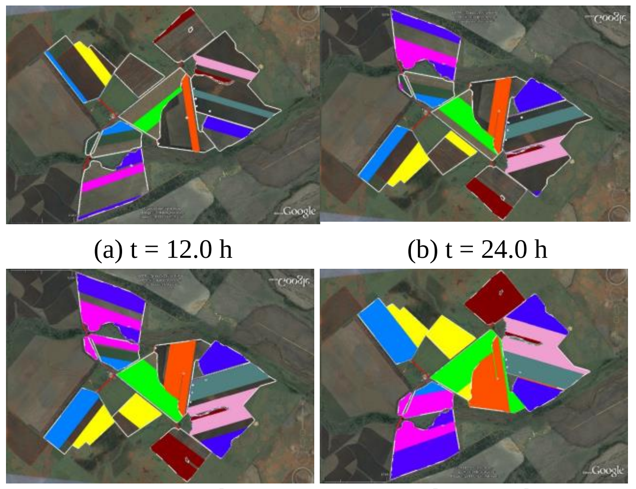

(a) t = 12.0 h (b) t = 24.0 h
(c) t = 36.0 h (d) t = 48 h

Figure 7: Integrated coverage with ten robots at four time points. The elapsed times are shown next to the figure identifier.

robot covers an equal area of the farm, but multiple fields are covered simultaneously as expected.

Summary statistics are shown in Table 2. Performance in the integrated case is marginally better in all metrics. As expected, this improvement is due to reduced travel between fields. In the sequential case, all robots incur the cost of moving from field to field, whereas in the integrated case a given robot may operate primarily in a single field without the need for inter-field travel.

In practice, it may not be economically feasible to perform fully integrated coverage due to the cost of building and maintaining many refill stations. Further, fields may have different timing requirements due to the type of crop planted and the type and maturity of weeds that must be controlled. However, the marginally superior performance of integrated coverage motivates grouping fields to the greatest extent possible, and also motivates further work in optimising the number and placement of refill stations that is outside the scope of the present paper.

4.2 Field trial experimental setup

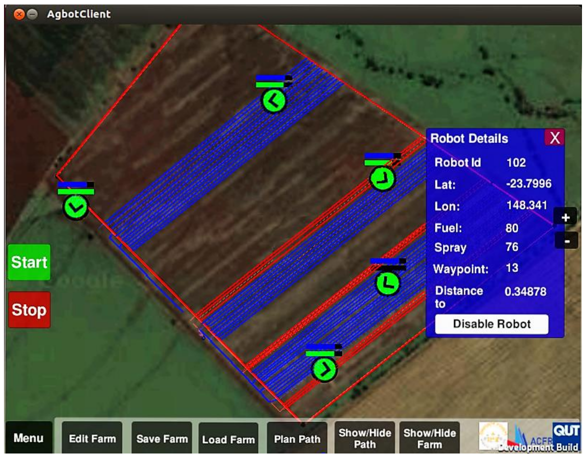
The scenario used for the experiment consisted of four simulated robots and two robots which operated over a 59-ha field. In this experiment the user interface was run in parallel with the Unity3D server and the coverage planner on a laptop running Ubuntu 12.04.

The GUI was used to configure the boundary polygon of the field, define row orientation, input roads and the refilling stations. The user interface is shown in Figure 9.

Throughout the duration of the experiment, robots were monitored via the user interface. Furthermore all robots were able to correctly respond to start and stop commands issued through the interface. During the experiment the robot docked five times. One of the robots reset several times during the experiments requiring the system to send a new plan, accounting for where the robot had already covered.

Table 2 Comparison of integrated and sequential coverage of ten fields in simulation. The robots travel at 5 km/h with 5 m coverage width. In both cases, the number of robots is 10 and the total area covered is 1145 ha. Means shown are per robot, with standard deviations shown in parentheses.

	Integrated coverage	**Sequential coverage**
Mean area covered (ha)	114.5 (3.69)	114.5 (5.9)
Mean distance travelled (km)	239.9 (1.4)	262.0 (5.7)
Mean time (h)	48.0 (0.3)	52.5 (1.1)
Mean coverage rate (ha/h)	2.4 (0.1)	2.2 (0.1)



Robot Details

Robot Id	102
Lat:	-23.7996
Lon:	148.341
Fuel:	80
Spray	76
Waypoint:	13
Distance to	0.34878

Figure 9. This screenshot shows the user interface during field trails each green circle represents an active robot. Each bar above the robot shows the battery (blue) and spray (green) levels. Completed paths are shown in red and future paths in blue. The 6th robot in this scene has been switched off and therefore doesn't appear on the interface. The blue panel shows detailed information about robot id 102.

5 Conclusions

Robotics has the potential to help with some of the challenges facing farmers. In one possible solution, large farm machinery may be replaced with numerous small robots. Hence, to alleviate extra workload on the farmer this paper has described a user interface and multi-robot multi-paddock coverage planner suitable for agricultural robotics.

During field trials the GUI was able to provide the user with an interactive Google maps image that was used to configure field boundaries, roads and refuelling stations. The coverage planner was then able to take the defined geo-referenced locations and create a multi robot coverage plan. The GUI was then able to provide the user with a visual representation of the robots current state and plans.

This paper also presented further research into sequential and integrated coverage approaches using teams of varying sizes. Using a simulated environment of 1145ha it was indicated that a team size of ten robots is practical for weed control in this typical application scenario and approximates the time required for current operation. Integrated coverage marginally outperforms sequential coverage, but this benefit comes at the expense of multiple refill stations which may be cost-prohibitive.

An important avenue for future work is to optimise the placement of refill stations to reduce travel time inefficiencies. It is also interesting to consider when and where to move a refill station during coverage, and to coordinate robot motion accordingly. Refueling the robot, either by recharging the battery or refilling a petrol or diesel fuel tank, in addition to spray tank refilling is an interesting question left for future work.

6 Acknowledgements

This work was supported in part by the Australian Research Council Linkage Project LP110200375 "Robotics for Zero Tillage Agriculture" awarded to the Queensland University of Technology, Swarm Farm Robotics and the Australian Centre for Field Robotics; the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program; the Australian Centre for Field Robotics; and the New South Wales State Government.

7 References

[Ball *et al.*, 2013] David Ball, Patrick Ross, Andrew English, Tim Patten, Ben Upcroft, Robert Fitch, Salah Sukkarieh, Gordon Wyeth, and Peter Corke. Robotics for sustainable broad-acre agriculture. *9th conference on Field and Service Robotics*, Brisbane, Australia, 2013.

[Blackmore *et al.*, 2001] Simon Blackmore, Henrik Have, and Spyridon Fountas. A specification of behavioural requirements for an autonomous tractor. *6th International Symposium on Fruit, Nut and Vegetable Production Engineering conference.* Potsdam, Bornim, Germany, 2001.

[Bochtis and Sørensen, 2009] D D Bochtis, and C G Sørensen. The vehicle routing problem in field logistics part I. *Biosystems Engineering* 104: 447-457, 2009.

[Bochtis and Sørensen, 2010] D D Bochtis, and C G Sørensen. The vehicle routing problem in field logistics: Part II. *Biosystems Engineering* 105: 180-188, 2010.

[Choset, 2001] Howie Choset. Coverage for robotics–A survey of recent results. *Annals of mathematics and artificial intelligence*: 113-126, 2001.

[Choset, 2005] Howie Choset. Principles of Robot Motion: Theory, Algorithms, and Implementation. Cambridge, MA, MIT Press, 2005.

[Galceran and Carreras, 2013] Enric Galceran, and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 2013.

[Hameed, 2013] I. A. Hameed. Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. *Journal of Intelligent & Robotic Systems*: 1-19, 2013.

[Hamza and Anderson, 2005] M.A Hamza, and W.K. Anderson. Soil compaction in cropping systems: A review of the nature, causes and possible solutions. *Soil and Tillage Research*: 141-145, 2005.

[Heap, 2014] Ian Heap. Global perspective of herbicide-resistant weeds. *Pest Management Science*: 1306-1315, 2014.

[Jin and Tang, 2011] Jian Jin, and Lie Tang. Coverage path planning on three-dimensional terrain for arable farming. *Journal of Field Robotics* 28: 424-440, 2011.

[Laengle and T.Hoeniger, 1997] T. Laengle, and T.Hoeniger. Cooperation in human-robot-teams. *Informatics and Control*, St Petersburg, 1997.

[Oksanen and Visala, 2009] Timo Oksanen, and Arto Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics* 26: 651-668, 2009.

[Pedersen *et al.*, 2006] S M Pedersen, S Fountas, H Have, and B S Blackmore. Agricultural robots—system analysis and economic feasibility. *Precision agriculture* 7: 295-308, 2006.

[Rekleitis *et al.*, 2008] Ioannis Rekleitis, Ai Peng New, Edward Samuel Rankin, and Howie Choset. Efficient boustrophedon multi-robot coverage: an algorithmic approach. *Annals of Mathematics and Artificial Intelligence* 52(2-4): 109-142, 2008.

[Xu *et al.*, 2014] Anqi Xu, Chatavut Viriyasuthee, and Ioannis Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots* 36(4): 365-381, 2014.