Vision-based Obstacle Detection and Navigation for an Agricultural Robot

David Ball, Ben Upcroft, Gordon Wyeth, and Peter Corke
School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane QLD 4001, Australia and ARC Centre of Excellence for Robotic Vision, Australia
Andrew English and Patrick Ross
School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane QLD 4001, Australia
Tim Patten, Robert Fitch, and Salah Sukkarieh
Australian Centre for Field Robotics, The University of Sydney, Sydney NSW 2006, Australia
Andrew Bate
SwarmFarm Robotics, Queensland, Australia

Received 31 March 2015; accepted 29 October 2015

This paper describes a vision-based obstacle detection and navigation system for use as part of a robotic solution for the sustainable intensification of broad-acre agriculture. To be cost-effective, the robotics solution must be competitive with current human-driven farm machinery. Significant costs are in high-end localization and obstacle detection sensors. Our system demonstrates a combination of an inexpensive global positioning system and inertial navigation system with vision for localization and a single stereo vision system for obstacle detection. The paper describes the design of the robot, including detailed descriptions of three key parts of the system: novelty-based obstacle detection, visually-aided guidance, and a navigation system that generates collision-free kinematically feasible paths. The robot has seen extensive testing over numerous weeks of field trials during the day and night. The results in this paper pertain to one particular 3 h nighttime experiment in which the robot performed a coverage task and avoided obstacles. Additional results during the day demonstrate that the robot is able to continue operating during 5 min GPS outages by visually following crop rows.

1. INTRODUCTION

Farms are under constant pressure to become more productive to feed a growing population while also balancing long-term sustainability. Traditionally, farms have become more productive through increased mechanization and improvements in crop breeding and genetics along with higher-precision guidance systems. Blackmore, Have, and Fountas (2001) proposes a robotic solution for sustainable agriculture that increases productivity while reducing environmental impact. This solution uses coordinated teams of small and lightweight robots, which is similar to having a large human workforce that would provide individual plant treatment, without the high labor cost. An economic analysis demonstrates the potential for this approach, although the sensors form a significant cost for the robot platform (Pedersen, Fountas, Have, & Blackmore, 2006), and hence potentially a barrier for commercialization. As this approach relies on multiple robots, each robot must use low-cost sensors for obstacle detection and localization.

Direct correspondence to: David Ball: david.ball@qut.edu.au

This paper describes a system for inexpensive autonomous agricultural robots with a focus on two key innovations. The first is a new visually aided localization system using inexpensive global positioning system (GPS), inertial, and camera sensors that allow the robot to continue to follow crop rows during GPS outages. The second is a novelty-based obstacle detection system using a stereo vision sensor. The detection system works by learning a model of the typically uniform cropping fields; regions outside the model are novel and hence candidate obstacles. These two innovations form part of our complete navigation system, which generates collision-free kinematically feasible paths in large outdoor environments. This extends our previous work (Ball et al., 2013) experimentally in the following ways: performing the coverage and obstacle avoidance task at night, and doing experiments with visually aided localization in crop rows. This paper also has significantly expanded descriptions of the system components and a discussion of the results, including problems and lessons learned. An action shot of the robot avoiding obstacles can be seen in Figure 1.

Our one real robot has been tested for coverage tasks over several weeks of field trials. This paper presents results

Figure 1. Action shot of the autonomous robot (left) at night approaching two obstacles: a utility vehicle (middle) and a white tank (right). The bright strobe light is synchronized with the stereo camera pair to illuminate the scene. The red light is from a commercial weed detection system.

from one particular three-hour coverage task performed at night. (Run-time is limited by battery life.) Even using inexpensive GPS-INS sensors, the robot is able to localize well enough for the task. These results demonstrate that the robot can successfully avoid various obstacles that are typical in a farm environment, including drums, vehicles, branches, and people. Further results during the day demonstrate that this visually aided localization allows the robot to continue accurately to follow crop rows for artificial 5 min GPS outages.

The remainder of the paper is organized as follows. The following section reviews relevant literature in visual obstacle detection, crop row following, and path planning. Section 3 describes the design of the system beginning with an overview of the task and robot before describing each of the individual parts of the system in detail. Section 4 contains the experimental details and the results of the experiments. Section 5 contains a discussion of the results and the lessons that were learned.

2. LITERATURE

Commercially, advances in technology have paved the way for vehicle guidance in agriculture, enabling what is known as controlled traffic farming (Taylor, 1983). Vehicle guidance for agricultural machines have become more precise using GPS (Bell, 2000) and more robust using local techniques for row tracking (Billingsley & Schoenfisch, 1997; Reid & Searcy, 1987). However, these advances are not sufficient for autonomous operation as they lack a navigation and obstacle avoidance system. Ollis and Stentz (1997) describe early work in developing autonomous farm machinery using vision to detect the boundary between cut and uncut crop and novelty-based obstacle detection. This system was demonstrated harvesting 48 hectares of crop (Pilarski et al., 2002). In other work, a neural network detects obstacles in the path of the robot and waits for human advice on each detected obstacle (Stentz, Dima, Wellington, Herman, & Stager, 2002). For early research in Japan, see Torii et al.'s (Torii, 2000) review of techniques, including using neural networks, genetic algorithms, and fuzzy logic for robots capable of tillage, planting, and plant care. The following three sections review literature relevant to the core parts of our robot design.

2.1. Obstacle Detection

Previous work in the field of visual obstacle detection can be broadly separated into two groups. The first is based on the structure of the scene from stereo matching or similar (Hadsell et al., 2009), and the second is based on monocular classification strategies that use prior knowledge about types of terrain and obstacles present (Angelova, Matthies, Helmick, & Perona, 2007; Filitchkin & Byl, 2012; Khan, Komma, & Zell, 2011; Vernaza, Taskar, & Lee, 2008).

Agricultural fields consist predominately of vegetation such as crops, crop residue (stubble), and weeds. Pure scene structure methods can suffer from the inability to distinguish between traversable and nontraversable objects in the scene, e.g., the difference between a rock and dense vegetation; the latter may be considered partially traversable by a sufficiently large vehicle. Additionally, the accuracy of any method based on stereo matching is limited by the accuracy of the matching itself. Stereo matching accuracy quickly decreases at long range (Hadsell et al., 2009), making structural methods based on stereo vision typically limited to short range. There are solutions to increase the effective range of stereo matching using multibaseline cameras (Broggi et al., 2010; Milella & Reina, 2014), but there will always be a limit to the accuracy of structural methods.

Monocular classification strategies typically can extract semantic information from the scene to make decisions about traversability. Most classification strategies are limited by the descriptors, in that these descriptors exhibit

varying levels of change under different lighting conditions (Ranganathan, Matsumoto, & Ilstrup, 2013; Ross et al., 2013; Valgren & Lilienthal, 2007), making it difficult to generalize learning for longer-term experiments (Kim, Sun, Oh, Rehg, & Bobick, 2006).

Some authors have circumvented this issue by continually taking classes online with "near-to-far" learning (Dahlkamp, Kaehler, Stavens, Thrun, & Bradski, 2006; Hadsell et al., 2009; Reina & Milella, 2012). In this strategy, stereo or laser structure information is used to classify at close range, and monocular classification strategies are used to extend this to long range. Since the classifiers are trained on only recent data, they are significantly more robust to lighting changes.

To overcome the individual failure modes of structure and appearance, multiple cues can be fused. This is typically achieved using expensive sensor suites (Wellington, Courville, & Stentz, 2005), requiring sensor calibration and registration, processes that must be continually repeated to ensure long-term operation as inter-sensor motion is unavoidable. In previous work (Ross et al., 2014), we demonstrated a system for obstacle detection that determines candidate obstacle regions in the image by looking for "novel" regions. Since the novelty is based only on recent image data, this method has little lighting dependence, and it requires no pretraining. Additionally, this processing step removes the vast majority of stubble and crop from consideration as an obstacle, increasing the robustness of the system.

2.2. Crop Row Guidance

Methods to track the relative location of the crop rows that can augment or replace GPS have been researched for several decades; most early systems were based on monocular vision (Billingsley & Schoenfisch, 1997; Gerrish, Stockman, Mann, & Hu, 1986; Reid & Searcy, 1987). Closely related work involves tracking vehicle pose in orchards using scanning lasers (Libby & Kantor, 2011) or monocular vision (Zhang, Kantor, Bergerman, & Singh, 2012). There are two main methods of tracking the relative location of the crop rows: the first uses the three-dimensional structure of the crop rows, and the second uses appearance information from monocular cameras.

Methods that rely on 3D structure detect the height difference between the ground and crop rows using scanning lasers (Biber et al., 2012; Chateau et al., 2000; Fleischmann et al., 2013; Hiremath et al., 2014; Weiss & Biber, 2011), or stereo vision (Hanawa et al., 2012; Hiremath et al., 2014; Rovira et al., 2004; Rovira-Más et al., 2008). The use of 3D structure-based methods relies on detecting a height difference between the crop and the ground, thus they may not be applicable to very young or short crops.

Monocular appearance-based methods generally segment crop rows from soil, and then they find crop rows by fitting straight lines to the resulting binary image. Earlier examples mostly made use of monochrome cameras sensitive in the near-infrared spectrum (Åstrand & Baerveldt, 2005; Han, Zhang, Ni, & Reid, 2004; Marchant, 1996; Reid & Searcy, 1987; Tillett, Hague, & Miles, 2002) and segment plants from soil based on intensity since living plants tend to appear brighter than soil in the near-infrared spectrum. More recent methods generally use color cameras and segment plants from soil based on their green color (Bakker et al., 2008; Bossu et al., 2009; Ericson & Åstrand, 2010; Hiremath, Evert, Heijden, Braak, & Stein, 2012; Montalvo et al., 2012,2013; Romeo et al., 2012; Søgaard & Olsen 2003; Tu et al., 2014), which is more robust to illumination changes and shadows than near-infrared (NIR) approaches. The green from brown color segmentation approach is ineffective when treating fields in the period between harvest and planting (the fallow period) when the remaining crop residue (stubble) is no longer green, and segmenting based on color becomes difficult. Several approaches avoid the segmentation step by instead searching for periodic variations in image intensity (Hague & Tillett, 2001; Olsen, 1995; Søgaard & Olsen, 2003) under the assumption that the crop rows lie at the peaks in these periodic variations.

After segmenting crops from soil, monocular methods generally estimate the crop row positions by fitting lines to the segmented image. This is done using various methods, such as the Hough transform (Åstrand & Baerveldt, 2005; Bakker et al., 2008; Marchant, 1996; Sharifi et al., 2015), linear regression (Han et al., 2004; Søgaard & Olsen, 2003), fixed template matching (Tillett et al., 2002), or random sample consensus (RANSAC) (Halmetschlager et al., 2014). Fitting straight lines to binary segmented images of green plants is error-prone in fields with high weed loading since "incorrect" lines may be fitted. Tu et al. (2014) fit a flexible four-sided polygon around each crop row rather than fitting lines, while Bossu et al. (2009) avoided fitting lines by applying frequency analysis using the wavelet transform to identify crop rows.

Most monocular methods for tracking crop rows require crop- and field-specific details such as row spacing, row width, plant color, and soil color. Hiremath et al. (2012) reduced this requirement significantly by learning row spacing and row width online as part of the state vector of a particle filter. The probabilistic framework also includes wheel odometry to improve robustness in noisy environments. Halmetschlager et al. (2014) also applied a particle filter to a combined stereo vision and near-infrared method of tracking crop rows, and they found improved results over RANSAC for detecting crop rows.

3. SYSTEM DESIGN

This section describes the field environment, the robot (including the physical and software robot platform), and the algorithmic details of our navigation system.

Figure 2. The robot shown in a typical Australian broad-acre farm environment. The field has knee-high sorghum stubble in straight and parallel rows. An example of a wheel track that has caused hyper-compacted soil can be seen running from the bottom right to the middle left of the image.

The target environment for this robot is broad-acre crop fields. These fields are large, on the order of hundreds of hectares, and they consist of mostly loose soil. The rows of stubble (crop residue) are generally straight and parallel except for the headland turning regions at the ends of the rows. The farm environment can be seen in Figure 2. The distance between stubble rows is from 0.25 to 1.0 m, and the stubble is between 0.3 and 0.6 m high. The fields are generally flat except for a few gentle contour banks.

There has been a strong trend away from mechanical weed destruction by tilling the entire field toward increased use of herbicide to manage weeds—zero-tillage agriculture. However, this increase has led to highly resistant weeds, resulting in an unsustainable sole solution to weed management. Resistant weeds cost Australia alone approximately $4 billion per annum (Sinden et al., 2004). Therefore, the primary task envisaged for our robot is weed management, which in this case requires the weeding implement to cover the entire field exactly once. In this specific case, the robot has a commercial selective spray system that will detect green plants, which in fallow fields will be weeds. However, other implements for seeding, fertilizing, and harvesting are possible with the robot described in this paper. The overall system allows each robot to operate autonomously within an assigned section of the field without regular communication with the multirobot coverage system.

3.1. Robot Platform

The robot is based on an Ackermann-steered electric John Deere TE Gator, as shown in Figure 3. We used a RoPro Design interface to the vehicle that allows for autonomous computer control. The robot is approximately 3.2 m long, and it has a 5-m-wide spray boom implement and a 200 L spray tank. In an emergency, the robot can be stopped with a spring-applied pneumatic release emergency brake using one of five local physical buttons or a wireless estop button.

The sensors on the robot were chosen to be relatively low-cost since our approach relies on multiple inexpensive small machines. The robot has the following sensors:

- A forward-facing stereo pair of iDS UII-5240CP power over ethernet GigE cameras with wide-angle lenses (~US$1,400 each).
- Two quadrature encoders S63B-37ADQ-OCCP4-AF mounted on the rear wheels providing a resolution of 6 mm (~US$250 each).
- A CH Robotics UM6 inertial measurement unit (IMU) (~US$200).
- A Skytraq S1315F-RAW GNSS with a ceramic patch antenna (~US$350).

The robot has two onboard networks, namely a controller area network (CAN) for controlling the vehicle's

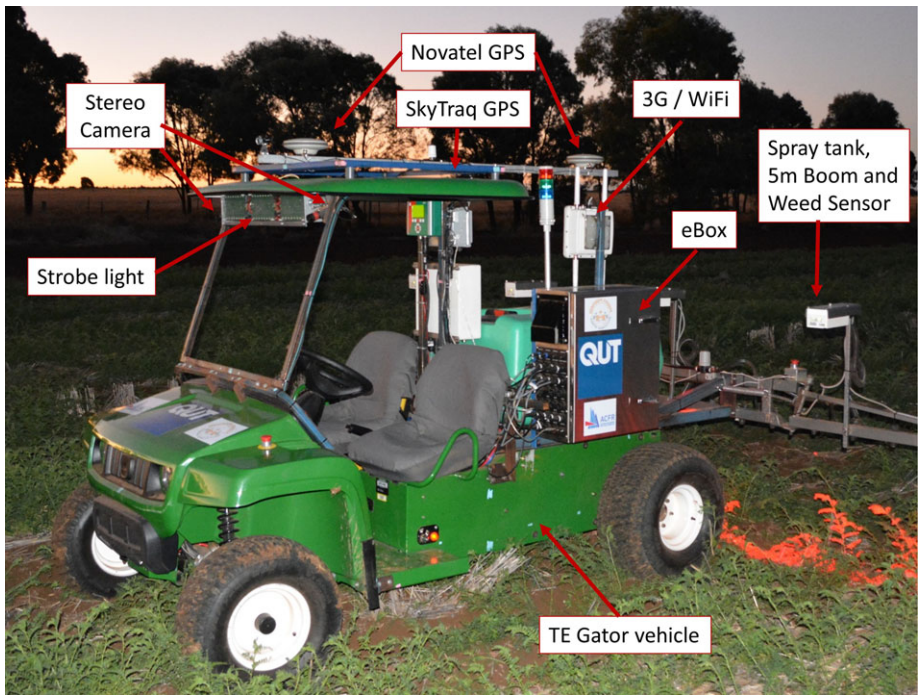

Figure 3. The agricultural robot platform. The base vehicle is an electric John Deere TE gator fitted with a RoPro Design conversion kit for conversion into a robot.

drive system, and a power over ethernet (POE) network for sensors. The robot also has a 3 G mobile interface to access the internet for communication with the multirobot planner.

The robot receives real-time kinematic (RTK) precision correction data from the SmartNet Australian Continually Operating Reference Station (CORS) network over a mobile data network. The corrections are passed to our robot's localization system (described in Section 3.4) and to a high-performance RTK GPS-INS system [a dual antenna Novatel FlexPack with Tactical Grade IMU, and the GPS specifies a horizontal accuracy of 0.01 m $\pm$ 1 ppm root mean square (RMS)] used only for ground truth for results.

3.2. Software and Hardware Details

The robot has two PCs running Ubuntu 12.04 and the robot operating system (ROS) Fuerte (Quigley et al., 2009). One PC runs the novelty-based visual obstacle detection algorithms described in Section 3.3, while the other runs nodes for navigation, vehicle control, and localization as described in Section 3.4. The two PCs communicate over the ethernet via a dedicated switch and router. One PC runs the ROS *master*, which provides the naming and registration services to all nodes. Therefore, nodes running on either PC can locate each other. After locating each other, the nodes then communicate directly. The nodes running on the robot and base station are shown in Figure 4 and communicate on topics using predefined messages. Most of the nodes are configured to output messages at 10 Hz.

For nighttime operation, the area in front of the vehicle is illuminated by a custom-designed strobe light mounted between and slightly behind the stereo camera pair. The pulsing light is synchronized to the stereo camera shutters via a trigger line, and it was designed to provide adequate illumination up to 10 m in front of the vehicle. Energy stored in a large capacitor bank drives 48 Cree XML EasyWhite 13 Watt LEDs pulsed at up to 2.5 times their rated continuous DC current to output a luminous intensity of up to 28,000 candela. For these experiments, the light operated at 10 Hz with a 4 ms pulse width. The light is configured over the ethernet via a ROS driver. The pulsed light enables higher light levels and a reduced average current consumption compared to a light source that is always on.

The stereo cameras were calibrated using the Automatic Multi-Camera Calibration (AMCC) MATLAB calibration toolbox (Warren, McKinnon, & Upcroft, 2013).

3.3. Novelty-based Obstacle Detection Using Stereo Vision

This section describes the novelty-based obstacle detection system, motivated by the observation that the typical appearance of the environment is uniform, and obstacles typically have a different appearance. A novelty detector identifies regions of the image that deviate from the learned

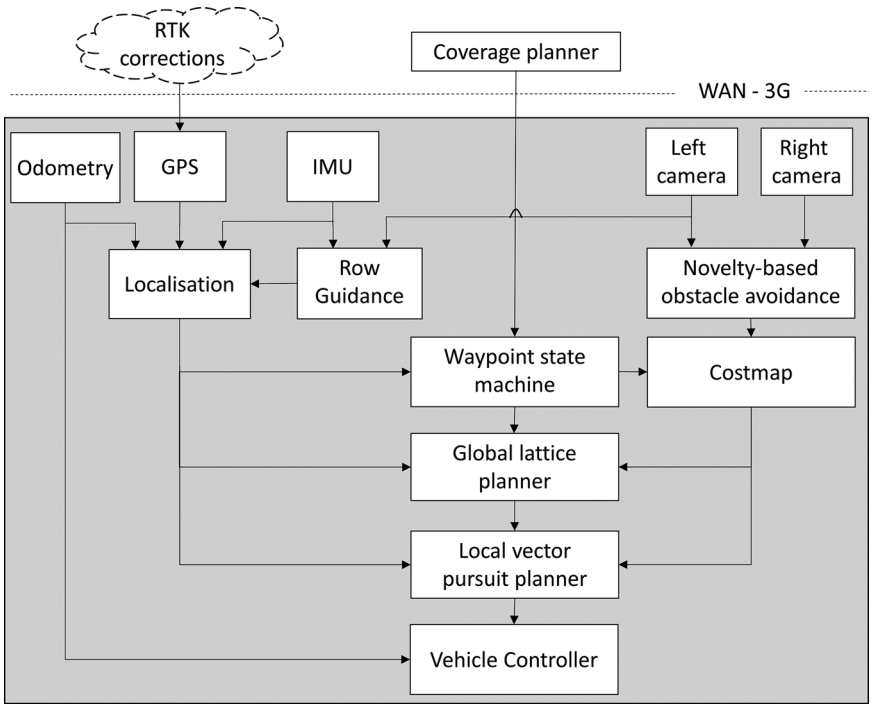

Figure 4. Schematic of the robot software platform. The general structure is based on the ROS *move_base* framework.

model of the environmental appearance, and these candidate obstacles are confirmed using stereo matching.

Figure 5 shows an overview of the novelty system. Images are first processed to detect novel regions. Stereo matching is performed on these novel regions to generate a 3D point cloud. The point cloud is then filtered to find the obstacles. As the model update step takes several seconds, the model is updated asynchronously to the rest of the system. To prevent detected obstacles from being incorporated into the model, an obstacle mask is sent to the model. The output of the system is a 2D local obstacle map for path planning and a mask of the locations of the obstacles in image space. This mask is used to ensure that obstacles are not added to the typical model of the environment, and as such they are always considered to be novel.

3.3.1. Image Descriptors

The image is split into an 80×64 grid, each cell of which is 16×16 pixels in size. The descriptor for each of these cells is a 7D vector that comprises the location in image space, the average CIE Lab color, and a 2D texture descriptor. The components of the Lab color space describe the lightness (L) and the chrominance (a and b channels). This color space was chosen because the color space is perceptually uniform.

The texture descriptors are computed using the response of the grayscale image to the eight 3×3 Law's masks (Laws, 1980) excluding the low-pass mask. These are smoothed using a 3×3 box filter to produce eight smoothed response images, denoted $L^m,\ m \in 1, \ldots, 8$. The box filter is used for processing speed.

The first dimension of the texture descriptor is based on the texture strength:

$$S_{i,j} = \frac{\max_m L^m_{i,j}}{\frac{1}{8}\sum_m L^m_{i,j}}, \tag{1}$$

where i, j indicate the pixel location in the image. The second dimension of the texture descriptor is based on the texture response:

$$+\, R_{i,j} = \log\left(1 + \frac{1}{8}\sum_m L^m_{i,j}\right). \tag{2}$$

These values are combined into the descriptor by taking their average over the entire 16×16 cell given by

$$\hat{\boldsymbol{x}} = [L \quad a \quad b \quad u \quad v \quad S \quad R]^T, \tag{3}$$

where L, a, and b indicate the components of the Lab color space, u and v indicate the location of the center of the block in image space, and S and R indicate the average strength and response of the pixels inside the block. Each of the location and average color variables was scaled to be in the range $[0,\ 1]$. The texture cues were not rescaled since their range is ill-defined.

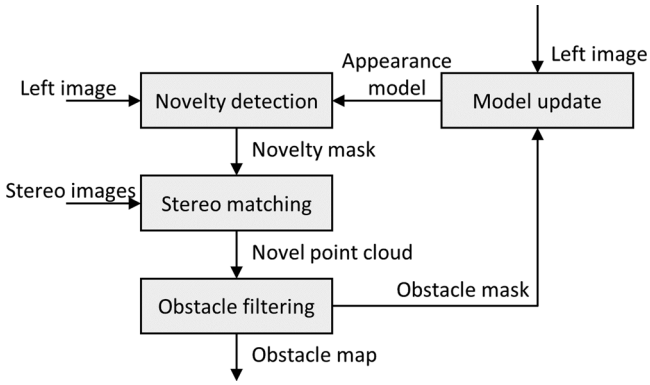

Figure 5. Overview of the system. The novelty mask is used to direct stereo matching. The resultant point cloud is filtered to produce an obstacle map, which is fed back into the model update node to prevent obstacles from being added to the model of the normal environmental appearance. Note that the model update is performed asynchronously to the rest of the processing pipeline.

3.3.2. Novelty Detection

Novelty is detected by estimating the probability density in descriptor space, and by looking for regions with low probability density. Due to the changing nature of the environment, standard Parzen windows are unsuitable as they provide the same weighting to old samples as to newer ones. We employ instead a variant of Parzen windows that allows the weighting of individual samples to be specified as

$$p(\boldsymbol{x}) \approx \frac{1}{\sum_i w_i} \sum_i w_i k_\sigma (\boldsymbol{x} - \boldsymbol{x}_i), \tag{4}$$

where $p(\boldsymbol{x})$ is the probability density at a query point $\boldsymbol{x}$, w_i is the weight of sample $\boldsymbol{x}_i$, and $k_\sigma(\boldsymbol{x})$ is the kernel function with a width parameter σ. This allows the weighting of samples to be gradually decreased over time so that new samples have more bearing on the result given by

$$w_i^n = (1 - \lambda)^n, \tag{5}$$

where λ is the forgetting factor, and n indicates the number of model update steps since the sample was added. A Gaussian kernel with spherical covariance is used such that

$$k_\sigma(\boldsymbol{x}) = \frac{1}{\sigma^D (2\pi)^{\frac{D}{2}}} \exp\left[-\frac{\boldsymbol{x}^T \boldsymbol{x}}{2\sigma^2}\right], \tag{6}$$

where D is the dimensionality of the descriptor. The descriptors are prescaled by a weighting vector $\boldsymbol{s}$ to bias some dimensions:

$$\boldsymbol{x} = \boldsymbol{s}^T \hat{\boldsymbol{x}}. \tag{7}$$

Queries are marked as novel if $p(\boldsymbol{x}) < \rho$, where ρ is some predefined novelty threshold. Once each cell has been labeled as novel or not, the result is passed through an islanding filter, which ensures that only novel regions that are connected to other novel regions are kept. The labeling is then dilated, which expands the novel regions. Note that these operations are performed at the cell level.

3.3.3. Obstacle Detection

Once novel image regions are identified, stereo matching is performed on these regions. In this work, we utilize LIBELAS (Geiger, Roser, & Urtasun, 2011), which was found empirically to give superior results when compared with other stereo matching algorithms for online operation.

This produces a point cloud for only the novel image regions. These points are filtered based on their height and distance from the camera. If both of these fall into a predefined range, then the point is labeled as an obstacle. This assumes that the ground is locally flat and the robot is flat on the ground, which is an appropriate assumption for broadacre field environments.

3.4. Visually Aided Localization

Precise navigation in cropping fields with GPS is becoming standard practice in the agricultural industry. Large manned machinery is steered automatically along fixed predefined tracks. However, GPS suffers from occasional outages in position due to communication link failures and loss of satellite lock due to occlusion by obstacles such as trees. Precise field operations, such as driving between crop rows, typically require costly RTK-GPS receivers along with nearby infrastructure to compute differential corrections, along with a permanent data link. For expensive conventional farm machines, a high-cost GPS-INS system may be justified; however, with our goal of numerous inexpensive robots, it is critical to reduce the cost of the localization system.

Our approach to robot localization is to fuse data from many lower-cost sensors together to produce a localization system that is robust to temporary failures in one or more

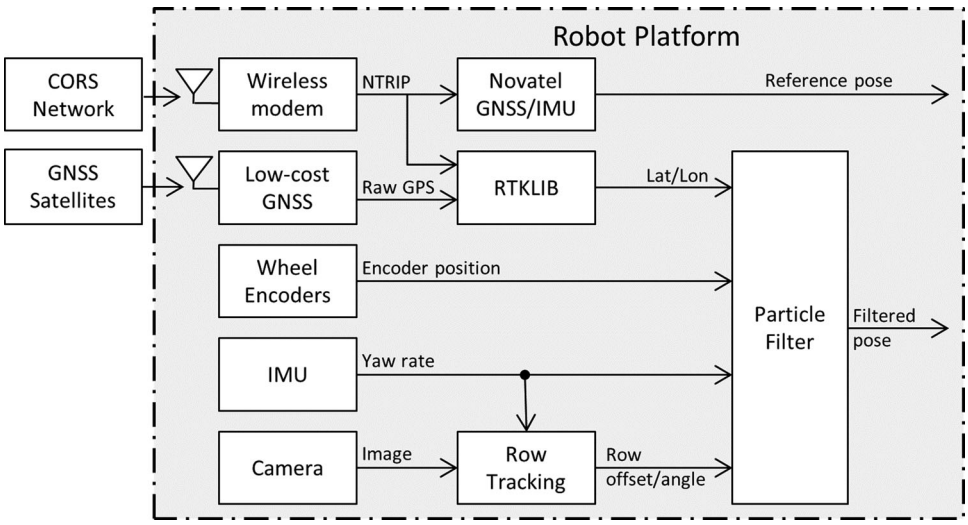

Figure 6. Overview of the localization system. The wireless modem operates on the 3 G mobile date network. Networked transport of RTCM via internet protocol (NTRIP) is a protocol for streaming differential GPS.

of the sensors. We use a particle filter to fuse GPS, wheel odometry, IMU, as well as our visual row-tracking method. This strategy allows our localization system to navigate for extended periods of time without one or more of the information sources available. This is particularly useful for bridging outages in GPS due to interruptions in the RTK corrections data link as well as a loss of GPS satellite signal due to occlusions from trees. Multiple redundant navigation sensors also allow us to select lower-cost sensors that may have lower availability. For example, we use a consumer-grade single-channel GPS that has a longer reacquisition time than more expensive dual-channel receivers, and we use a low-cost IMU that is capable of dead reckoning for only short periods of time. Figure 6 shows an overview of the localization system.

RTK corrections are received via 3 G cellular connection from the local Continuously Operating Reference System (CORS) network SmartNet Australia. Rather than using the GPS position calculated by the GPS chipset, a position is calculated in software on the navigation computer by combining raw GPS measurements with the RTK corrections using the freely available open source package RTKLIB (Takasu & Yasuda, 2009). This enables inexpensive RTK-GPS costing hundreds of dollars rather than the thousands of dollars typical of most RTK-GPS systems. Under ideal conditions, low-cost receivers using RTKLIB can achieve accuracies of several centimeters (Jensen, Larsen, Simonsen, & Jørgensen, 2012; Osterman, Godeša, & Hočevar, 2013; Stempfhuber & Buchholz, 2011; Takai et al., 2011). However, in our previous work (English et al., 2013) we found that in rough terrain with a low-cost antenna, position solutions typically contain a slowly varying bias of up to a few tens of centimeters. Additionally, when GPS reception is interrupted due to obstacles or loss of correction data link, single-frequency (L1) GPS receivers often take several minutes to reacquire an accurate RTK fix. This lower availability makes visual crop row tracking a valuable addition that enables navigation to continue during these periods.

The particle filter localization system presented in this work builds off of our previous work (Ball et al., 2013) by including our visual crop row tracking algorithm (English, Ross, Ball, & Corke, 2014) as an input to the particle filter to greatly reduce the position drift of the vehicle during GPS outages.

3.4.1. Visual Row Tracking

Crop rows are a common feature in most cropping fields, particularly in zero-tillage cropping where tilling the soil is avoided so that the crop residue is left standing year round to reduce soil erosion. This leaves a visible row pattern that can be tracked using computer vision to augment GPS navigation. Most visual methods for crop row tracking use color-segmentation techniques to track green crop rows; however, visually tracking crop rows in the fallow period when there is only dead crop residue has remained a challenge.

In previous work (English et al., 2014), we presented a vision-based row tracking method that enables visually aided navigation in these fields where the crop rows are challenging to detect. This method eliminates crop-specific details such as color, spacing, and periodicity by extracting and tracking the direction and lateral offset of the dominant parallel texture in a simulated overhead view of the scene. Previous results demonstrated that the method is able to guide a robot in a wide variety of fields, both day and night,

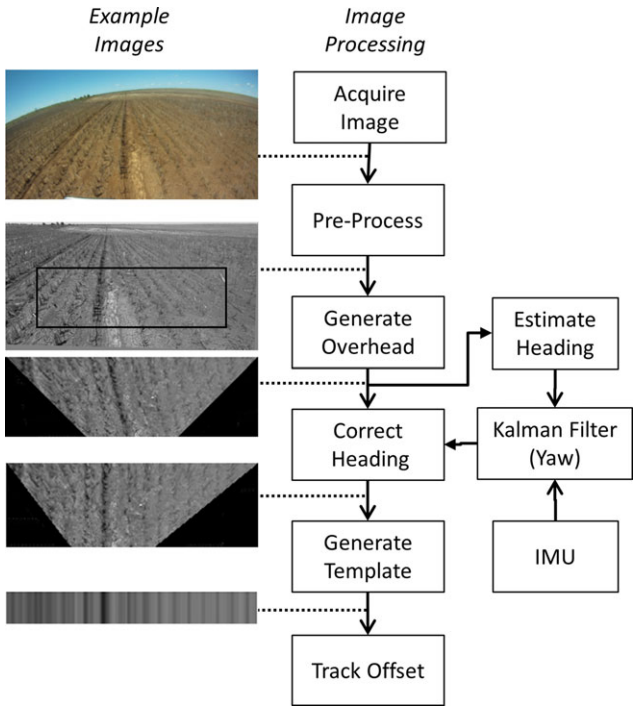

Figure 7. Block diagram of the algorithm along with example images.

as well as in very difficult conditions such as fields with only dead crop residue (stubble). In this work, we use a slightly modified version of this visual crop row tracking method to track crop rows. The details of this method are summarized here for clarity. We also integrate crop row tracking measurements into the robot's navigation particle filter as explained in Section 3.4.2.

The crop row tracking method uses a single forward-facing camera (in this case the left camera in the robot's stereo pair) to track both the heading of the vehicle with respect to the crop, as well as the frame-to-frame lateral offset of the vehicle with respect to the crop rows. To eliminate crop-specific parameters such as color, spacing, height, and periodicity, the crop rows are modeled as an arbitrary planar parallel texture.

The main steps in the algorithm are shown in Figure 7 and are listed here. For more specific implementation details, refer to English et al. (2014).

- Preprocess the image to correct for lens distortion, convert to grayscale, and downsample to improve processing speed.
- Warp a rectangular region of interest in the preprocessed view to a synthetic overhead view using planar homography based on the camera pitch.
- Estimate the vehicle's heading relative to the crop rows by estimating the direction of the dominant parallel texture in the overhead image. A Kalman filter fuses this heading estimate with the yaw rate from the IMU for a more robust heading measurement.
- Correct for heading in the overhead view by skewing the image using the estimated heading.
- Generate a "frame template" by summing the columns of the skewed images.
- Estimate the vehicle's motion perpendicular to the crop rows by searching for the sideways offset by comparing the current frame template to the previous frame template using zero-mean normalized cross correlation (ZNCC) with subpixel interpolation.

The abstraction of crop rows to an arbitrary planar parallel texture enables virtually parameter-free estimation of the vehicle's movement perpendicular to the crop rows, even in fields where the rows are difficult to see. For example, when there are no green crop rows but only dead crop residue from the previously harvested crop, segmenting the crop rows from the ground can be very difficult and error-prone. The abstraction of crop rows eliminates those difficulties. It also allows other texture-based cues that are not necessarily from a crop, such as the shadow of the crop or wheel tracks to assist in navigation. A disadvantage of this abstraction is that since we do not explicitly identify the crop, we cannot use the system to guide the robot to drive with its wheels between two crop rows. However, driving in the rows is appropriate in some circumstances, such as in the fallow period, which is the case for the cropping field used in our experiments. To extend functionality in order to prevent the wheels from damaging live crops, the individual crop rows must be identified.

There may not always be a viable crop signal to track, e.g., at the ends of the fields. To avoid returning erroneous results, we use a measure of the strength of the parallel texture to determine if the scene contains crop rows for reliable tracking. This strength measure is used automatically to switch off the crop row tracking in regions such as bare patches and at the ends of the field where there is either no crop or the crop has been planted in two directions leaving no clear parallel texture to track. Using crop row tracking only when there is sufficient parallel texture greatly improves the robustness of the system.

The implementation of crop row tracking used in this work contains some simplifications compared to English et al. (2013). Specifically, in the previous work we used our estimated camera pitch and roll from an IMU or visual horizon tracking to correct our simulated overhead view due to the vehicle driving on uneven ground. This step has been omitted from this work since we found in practice that our crop row tracking method is robust to small amounts of vehicle roll and pitch. In fact, our low-cost IMU was not always able to reliably estimate roll and pitch to a sufficient degree, and there were difficulties in synchronizing the camera and IMU data streams. Additionally, we previously

calculated lateral movement by comparing the current frame template to the first frame template since tracking began. This has been changed to comparing to only the previous frame template since this provides greater robustness to changes in the visual appearance of the crop along the crop row. The disadvantage of this change is that frame-to-frame tracking introduces drift into the tracking, since small errors in each frame-to-frame comparison accumulate over time. However, in practice, the lateral drift is small in comparison to forward motion. This small drift was an acceptable compromise for the significantly improved robustness to changes in crop row appearance.

3.4.2. Particle Filtering

The localization system uses a particle filter that fuses the RTK-GPS position, wheel odometry, and yaw rate from the IMU as well as the offset and heading estimated by the vision-based crop row tracking. The particle filter represents the estimated probability distribution as a set of discrete weighted samples, and it allows for nonlinear process and measurement models as well as tracking of multiple state hypotheses.

In the filter formulation, the state vector is

$$\boldsymbol{x} = \begin{bmatrix} x_v \\ y_v \\ \theta_v \\ \varepsilon_{\dot{\theta}} \end{bmatrix}, \tag{8}$$

where x_v is the vehicle easting (m), y_v is the vehicle northing (m), θ_v is the heading of the vehicle (rad), and $\varepsilon_{\dot{\theta}}$ is the drift of the yaw rate gyroscope (rad/s).

3.4.3. Process Model

The process or motion model describes how the state vector evolves over time, that is, how each particle should be moved. The control inputs for the particle filter are

$$\boldsymbol{u} = \begin{bmatrix} v_A \\ \dot{\theta}_A \\ \tau_{\text{rows}} \\ a_{\text{rows}} \end{bmatrix}, \tag{9}$$

where v_A is the linear velocity derived from the wheel radius and sequential encoder positions (m/s), $\dot{\theta}_A$ is the rotational velocity measured by the IMU yaw rate (rad/s), τ_{rows} is the crop row tracking lateral offset from the previous frame (m), and a_{rows} is an "available" flag, which is 1 when row tracking information is valid and 0 when continuous row tracking has been lost.

The vehicle's motion is approximated at each iteration by a small movement forward followed by a small rotation in place. The vehicle's forward motion is determined by the velocity measured by the wheel odometry v_A:

$$x_{v,k+1} = x_{v,k} + v_{A,k}\Delta t \cos(\theta_{v,k}), \tag{10}$$

$$y_{v,k+1} = y_{v,k} + v_{A,k}\Delta t \sin(\theta_{v,k}). \tag{11}$$

The vehicle's rotation is determined either by the visual crop row tracking whenever available, or otherwise by the IMU yaw rate. Preference is given to the crop row tracking since it provides a much more accurate estimate.

When the crop row tracking is unavailable ($a_{\text{rows}} = 0$), such as at the ends of the field, the vehicle's update equation for vehicle rotation is

$$\theta_{v,k+1} = \theta_{v,k} + \dot{\theta}_{v,k}\,\Delta t. \tag{12}$$

When the vehicle's heading is updated by the crop row tracker output τ_{rows}, the update equation becomes more complex and requires examining some geometry, as shown in Figure 8.

The output of the crop row tracking algorithm τ_{rows} is defined as the frame-to-frame lateral offset of the crop row navigation point (point C in Figure 8) in a direction perpendicular to the crop rows (β_{rows}). At each update step, we determine the vehicle heading that moves the visual navigation point C perpendicular to the crop rows by a distance of τ_{rows} so that the visual navigation point C "follows" the output of the crop row tracking algorithm.

From Figure 8, we can see that the heading of the vehicle at time $k+1$ is

$$\theta_{v,k+1} = \beta_{\text{rows}} + \alpha, \tag{13}$$

where α is the angle of the vehicle with respect to the crop row angle β_{rows}. By applying trigonometry to the rightmost right-angle triangle in Figure 8 (highlighted in red), we observe that

$$\alpha = a\sin\left(\frac{\gamma + \tau_{\text{rows}}}{l_{AC}}\right), \tag{14}$$

where l_{AC} is the distance from the center of the rear axle at point A to the visual navigation point C. The unknown quantity γ is calculated using the leftmost right-angle triangle in Figure 8 (highlighted in blue):

$$\gamma = (l_{AC} - v_{A,k}\,\Delta t)\sin(\beta_{\text{rows}} - \theta_{v,k}). \tag{15}$$

Combining the equations in this section gives the final update equation for the vehicle,

$$x_{v,k+1} = x_{v,k} + v_{A,k}\,\Delta t \cos(\theta_{v,k}), \tag{16}$$

$$y_{v,k+1} = y_{v,k} + v_{A,k}\,\Delta t \sin(\theta_{v,k}), \tag{17}$$

$$\theta_{v,k+1} = \begin{cases} \beta_{\text{rows}} + a\sin\left(\frac{(l_{AC} - v_{A,k}\Delta t)\sin(\theta_{v,k} - \beta_{\text{rows}}) + \tau_{\text{rows}}}{l_{AC}}\right) & \text{if } a_{\text{rows}} = 1, \\ \theta_{v,k} + \dot{\theta}_{v,k}\Delta t & \text{otherwise.} \end{cases} \tag{18}$$

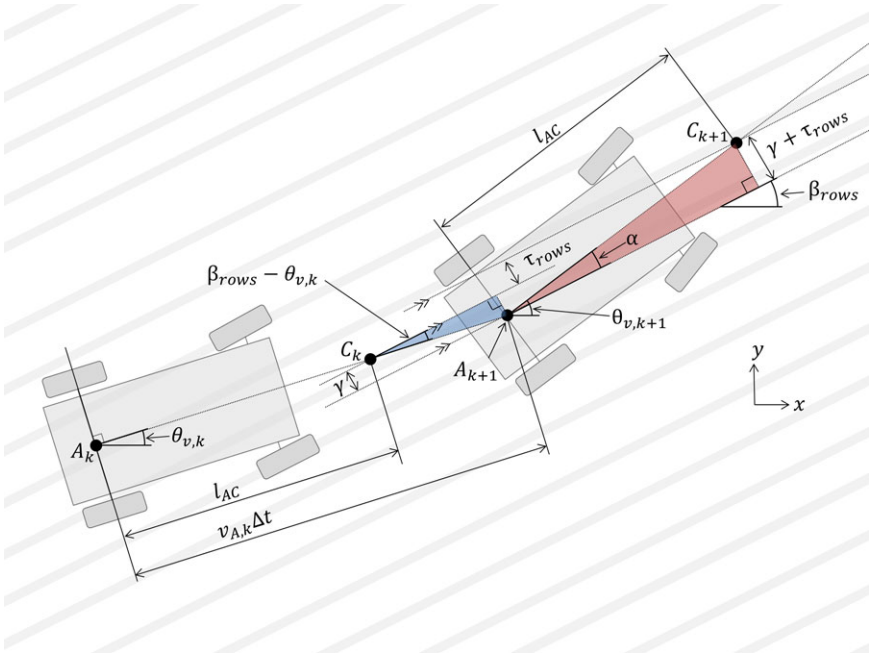

Figure 8. Motion model of the vehicle while using crop row tracking showing vehicle pose at time k and $k+1$. The vehicle motion is approximated by a forward movement by $v_{A,k}\Delta t$ followed by a rotation around the rear axle (A_{k+1}) to move the visual navigation point C to be a distance τ_{rows} from its previous crop row.

Note that we require knowledge of the crop row direction, τ_{rows}. This angle could be taken from a map of the crop rows, though in our test fields this map is trivial since the crop rows have all been planted in a single direction in straight and parallel passes. Straight and parallel crop rows planted with RTK-GPS guided tractors are now commonplace in broad-acre cropping.

3.4.4. Measurement Model

The measurement step of the particle filter involves reweighting particles based on the observations, followed by resampling the particles based on the position measured by the GPS. Our measurement model reweights particles based on GPS measurements. The measurement equations simply involve projecting GPS measurements taken at point G in Figure 9 back to the axle at location A given by

$$\boldsymbol{z} = \begin{bmatrix} x_{\text{GPS}} \\ y_{\text{GPS}} \\ \theta_v \\ \varepsilon_{\dot{\theta}} \end{bmatrix} = \begin{bmatrix} x_v + l_{ag}\cos\theta_v \\ y_v + l_{ag}\sin\theta_v \\ \theta_v \\ \varepsilon_{\dot{\theta}} \end{bmatrix} + \boldsymbol{v}, \qquad (19)$$

where the measurement noise $\boldsymbol{v}$ is modeled as independent Gaussian white noise.

The combination of RTK-GPS, visual crop row tracking, and odometry allows navigation to continue under a variety of failures and for the positioning estimation to degrade gracefully. When a good GPS lock is unavailable, navigation will effectively revert to visual crop row tracking. Since the visual crop row tracking provides estimates to the particle filter rather than directly controlling the vehicle's steering, the robot can continue to avoid obstacles when GPS is unavailable. Since the visual crop row tracking observes rows in a reasonably wide field of view, obstacles such as people

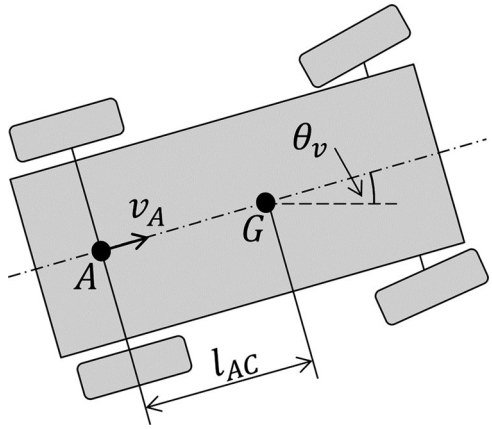

Figure 9. Vehicle model. The GPS measures the vehicle position at point G. The particle filter measures the vehicle position at the center of the rear axle at point A so the GPS position needs to be projected back to point A using the vehicle heading $\boldsymbol{\theta_v}$ and the distance $\boldsymbol{l_{AG}}$.

generally do not obscure the crop rows enough to interfere significantly with the visual crop row tracking. Also, larger obstacles that occlude the crop rows will generally cause the visual row tracking to disengage since few (if any) obstacles would look sufficiently like our model of crop rows (planar parallel texture with a vanishing point near the horizon). In the rare periods when both GPS and visual crop row tracking are unavailable, the vehicle's position estimation reverts to dead reckoning with the wheel velocity and IMU heading.

3.5. Coverage Planner

The task of weed management through controlled herbicide delivery is algorithmically an instance of the *coverage* problem (Choset, 2001). The goal of coverage is to plan a path such that the robot eventually visits (covers) all points within a defined area. Finding an optimal coverage path is related to the well-known traveling salesman problem (TSP), and it is known to be non-deterministic polynomial-time (NP)-hard (Galceran & Carreras, 2013). However, our application is a restricted case of coverage when robot motion is constrained to follow preexisting rows within a field. In this case, we do not need to solve the problem of choosing an optimal row orientation as in Oksanen & Visala (2009), but instead we must adapt existing general coverage algorithms.

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 is summarized 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, we have known row orientation. 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 two cases. (i) Within a cell, the path consists of straight line segments (rows) connected in a back-and-forth manner with short inter-row segments. (ii) Adjacent cells are also connected with short segments between adjacent rows.

3.6. Navigation and Vehicle Control

The robot's navigation system uses the ROS *move_base* architecture (Marder-Eppstein, Berger, Foote, Gerkey, & Konolige, 2010). The system is based on planning global and local paths through *costmaps*, which are standard components of the *move_base* architecture. See Figure 4 for a diagram of the architecture and Figure 10 for an example. The system has the following components, each of which is described in the following subsections.

- A waypoint state machine receives the complete list of paths from the multirobot coverage planner and sends these waypoints one at a time to the *move_base* global planner. It also generates a potential field, which is described later, to bias the robot onto the desired crop row.
- The costmap is generated from the vision system's obstacle map.
- The global planner plans long-term kinematically feasible paths to the desired waypoint through the costmap. This uses a lattice planner.
- The local planner generates the desired velocity commands to follow the global path while using the costmap to check for collisions. This uses a vector-pursuit technique.
- The vehicle controller sends the required commands to the vehicle to set the throttle, steering angle, and brake.

3.6.1. Waypoint State Machine

This node is responsible for breaking the path sent by the multirobot coverage planner into a sequential set of waypoints. It iteratively sends a goal waypoint to the navigation system and waits for completion. For rows, it also generates a potential field row funnel to bias path planning close to the desired row. The row funnel field is an inverted triangle of weights applied to the costmap, which biases the path generated by the global path planner. The funnel field strength is proportional to the distance from the row to be traversed by the robot. This node also continuously sends the position of the robot to the multirobot coverage planner.

3.6.2. Costmap

The costmap maintains a 2D representation of the environment near the robot, and it is a rolling 100×100 m map centered on the robot. The costmap is aligned to the direction of the crop rows, which allows the global planner to generate straight paths. This is because the global planner works in units the size and orientation of the costmap's grid cells, which are internally fixed when *move_base* starts. The costmap encodes the cost for occupying each 0.2×0.2 m cell. Obstacles are given high values to ensure that obstacles are avoided. The row funnel field is given low to medium values to bias the robot toward the desired row. The global planner uses this map to generate global paths, and the local planner uses it to check for collisions.

3.6.3. Global Planner

This module is responsible for planning kinematically feasible paths to reach the waypoint through the global costmap.

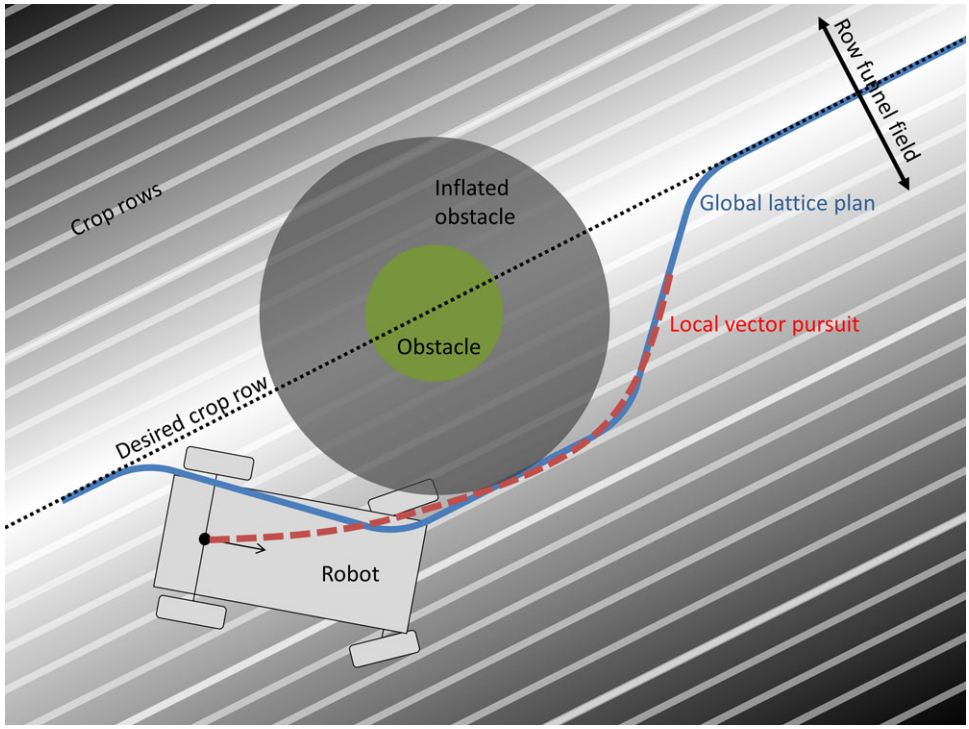

Figure 10. This diagram shows the different parts of the navigation system. Obstacles are added and inflated in the costmap. A row funnel biases the path selection toward the desired row. The global planner uses kinematically feasible motion segments to form a desired path. The local planner uses vector pursuit to minimize the error in the robot's distance to the path and its heading relative to the path.

The global planner will generate a new plan every ten seconds or if the local planner rejects the global plan. The local planner will reject the plan due to the presence of obstacles along the global path or if the robot deviates too far from the global path. The module uses the Search Based Lattice Planner (SBPL) (Cohen, Chitta, & Likhachev, 2010) to generate a path around obstacles to a goal location. This goal location is clipped to within the costmap.

Offline, a MATLAB tool was used to generate motion primitives that are kinematically feasible for the robot and the same size as the costmap cells. The key parameter for determining the motion primitives is the vehicles radius of curvature. The SBPL connects these motion primitives together by considering the cost of traversing the costmap cell and the cost of the motion. As described in the previous subsection, the costmap has the obstacles and the row funnel.

The global planner has more capability than is necessary for following straights rows. However, this planner provides flexibility for complex sets of obstacles and for future functionality requirements, such as moving between fields and docking.

3.6.4. Local Planner

This module determines the robot's immediate velocities to guide the robot onto the global path with the correct heading. A vector pursuit planner uses two hand-tuned proportional-integral controllers to minimize the error in the distance between the robot and the global path and between the robot's heading and the global path heading. The planner uses the robot's short-term predicted position as its reference, which is based on its current velocity.

The planner generates a prediction of the robot's future position by iterating the vector pursuit controller. This plan is used to check for collisions in the costmap. If there are collisions, then the local planner rejects the global plan and the global planner generates a new plan. As the robot has high inertia, and the global planner generates a new path relatively quickly (in approximately 1 s), the robot generally does not slow down noticeably between plans.

3.6.5. Vehicle Controller

The vehicle controller interfaces with the vehicle over CAN to several relays, encoder modules, analog outputs, and motors to control the robot's forward velocity, the

steering wheel angle, and the brake state. The controller runs a forward velocity control loop based on feedback from wheel encoders in order to set an analog throttle. The desired steering direction is sent over CAN to the steering wheel motor. The proportional-integral-feedforward gains were hand-tuned.

4. RESULTS

We performed two experiments to test the performance of the system, including a physical robot coverage task with obstacles and a row-following task with GPS outages. This section includes results for the obstacle detection system, the navigation system, and the localization system.

4.1. Coverage Experiment

In this experiment, the robot autonomously covered 6.3 hectares of cropping field at night over a period of approximately 3 h. This experiment tested the localization, obstacle detection, and path-planning systems. We placed several obstacles in the field, as shown in Figure 11, including a large hay bale, a utility vehicle, a 1,000 L herbicide tank, a person, some arranged branches, and two medium-sized drums. The tank was hidden behind the utility vehicle. The robot was commanded to perform parallel passes spaced 5 m apart along the length of the field to emulate spraying the field. The robot's position was calculated with the particle-filter localization system. There were no GPS outages in this experiment, and so visual crop row tracking was not used. Ground-truth position measurements were measured by the Novatel GNSS/INS system. In this particular coverage experiment, we did not spray liquid using the tank.

Data for these experiments were recorded on a single PC using the ROS tool *rosbag*. The messages have two time stamps, namely the time that *rosbag* receives the messages and the time in the message header when the driver or node processes the data. Later processing used the most appropriate time stamp. The two PC clocks were approximately synchronized using the Network Time Protocol (NTP) tool. Tables I and II contain the parameters that we used for these experiments.

Figure 11. Experimental setup showing the location of the obstacles for the coverage and obstacle avoidance experiment. The pictures of the obstacles are taken from the robot's camera stream. The red lines indicate the boundary for the experiment. The robot will cover the area by traversing back and forth seven times within this boundary. Note that the images have been manually brightened for the reader.

4.1.1. Localization Results

Figure 12 plots the localization error over the duration of the experiment, and Figure 13 displays the coverage of the robot's spray boom with the overlapped (black) and missed (red) areas. During this experiment, the particle filter was run with 2,000 particles. The process and measurement noise figures are shown in Table III. When performing a coverage task such as applying herbicide, the coverage tool (in this case the spray boom) should ideally cover the entire field once while minimizing overlap, which wastes resources such as time, energy, and herbicide. In this experiment, of the area assigned to the robot, 99.54% was covered by the spray boom while 8.77% of the field was covered more than once. These overlapped and missed areas include five deviations from the ideal "lawnmower" path due to avoiding obstacles placed in the field. These overlap and missed area statistics would be improved slightly were these obstacles not present. For the entire experiment, the RMS error in position output by the particle filter was 0.109 m when compared to the Novatel RTK GPS-INS system. (The description for the Novatel system states 0.01 m horizontal accuracy $\pm$ 1 PPM RMS.) The navigation sensors

Table I. Parameters for the navigation system.

Name	Value
Global costmap resolution	0.2 m
Global costmap size	100 × 100 m
Maximum range for an obstacle	9 m
Time allocated for global planning	2 s
Planner frequency	10 Hz
Recovery behaviors allowed	False
Max steering angle	0.78 rad
Wheelbase	1.77 m
Max linear velocity	1.3 m/s
Max rotational velocity	1.0 rad/s
Linear goal tolerance	2 m
Minimum turning radius	2.3 m
Time robot projected forward for vector pursuit planning	1.0 s
Vector pursuit distance gains	$K = 7, I = 0$
Vector pursuit angle gains	$K = 3, I = 0$
Row funnel width	20 m

Table II. Parameters for the obstacle detection system.

Name	Value
Novelty PDF threshold (ρ)	2×10^{-3}
Gaussian kernel width (σ)	0.1
Forgetting factor (λ)	10^{-3}
Sample minimum weight	10^{-5}
	[1]
	[1]
	[1]
Descriptor scaling vector ($\boldsymbol{s}$)	[2]
	[2]
	[2]
	[4]
Minimum obstacle height	−0.5 m
Maximum obstacle height	0.3 m
Maximum detection range	10 m

used in this experiment cost only hundreds of dollars, and yet these results demonstrate accuracies that are adequate for the task of spraying weeds.

4.1.2. Vision-based Obstacle Detection Results

Figure 14 shows a close-up of the region of the dataset that contained obstacles. No direct false positives were observed throughout the experiment; however, in several instances obstacles were inflated beyond their actual bounds. This is an artefact of the LIBELAS stereo matching algorithm.

The average image novelty over the entire dataset was less than 1%. These are mostly composed of small false positives in the novelty detector, which are correctly filtered by the stereo matching phase. There were some true positives from weeds for the novelty detector, which were still successfully filtered out since they were not high enough to be considered an obstacle. Figure 15 demonstrates an example in which LIBELAS extended an obstacle toward the viewpoint, effectively expanding it. This expansion was due to the proximity of the grass tuft to the vehicle as shown in Figure 16. LIBELAS interpolates into regions with little to no texture, and hence it connects these two points with a smooth surface. Since the novelty region extends somewhat beyond the boundary of the obstacles, part of this surface is considered a potential obstacle. This has the potential to occur only around actual obstacles typically, and hence it is not a critical failure; however, the virtual increase in the size of the obstacle leads to lower overall coverage, which is undesirable.

As discussed previously, stereo matching becomes unreliable at longer ranges, and so obstacles detected at longer ranges typically provided noisy or incorrect results. To deal with this issue, we introduced a maximum range of detection throughout all trials of 10 m, which was found empirically to effectively filter these out. This also helped to justify the locally flat world assumption since there should be less variation in the height of nearby obstacles. Despite this, we would like to extend the detection distance, since detection distances less than 10 m are close to the limit of the navigation system and vehicle reaction time.

The use of texture measures has increased the robustness of the novelty detector. However, in many situations it is still difficult to distinguish between obstacles and the ground terrain, such as when the obstacle is a hay bale. Figure 17 shows how the regions of the hay bale that are close to the ground are incorrectly labeled as non-novel, which caused delayed detections of the hay bale. This is a challenge for a novelty system that uses only monocular information.

When turning to avoid obstacles or at the end of the field, the appearance of the environment changes quite rapidly. During this time, the novelty detector has a significantly increased false positive rate, which leads to an increased probability of false positives making it through the system as shown in Figure 18. This issue cannot be addressed directly within the current framework, since it would require longer-term training to remember what these appearance changes look like. This solution reintroduces the problem of generalizing learning over changing lighting conditions.

The field of view of the system is quite limited (approximately 70 deg). Since the vehicle is quite wide, obstacles at the edge of the vehicle can only be seen when they are at least 4 m away, which provides a very small detection range given the maximum detection range of 10 m. We found that during forward motion, this was not a significant issue; however, when turning to avoid other obstacles,

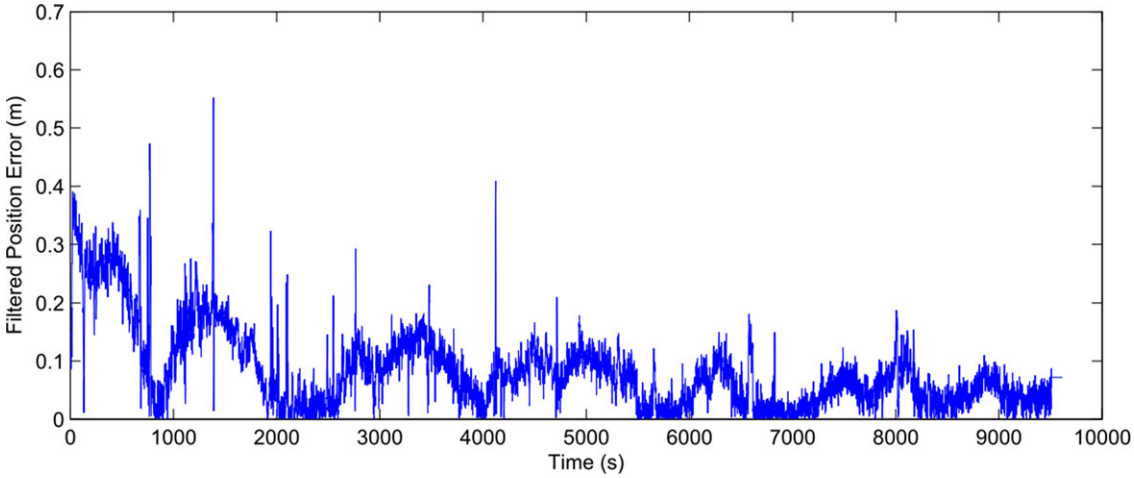

Figure 12. Absolute position error for the localization system during the coverage experiment. This error is calculated by comparing the global position given by our localization system to the global position from a midrange Novatel RTK GPS-INS.

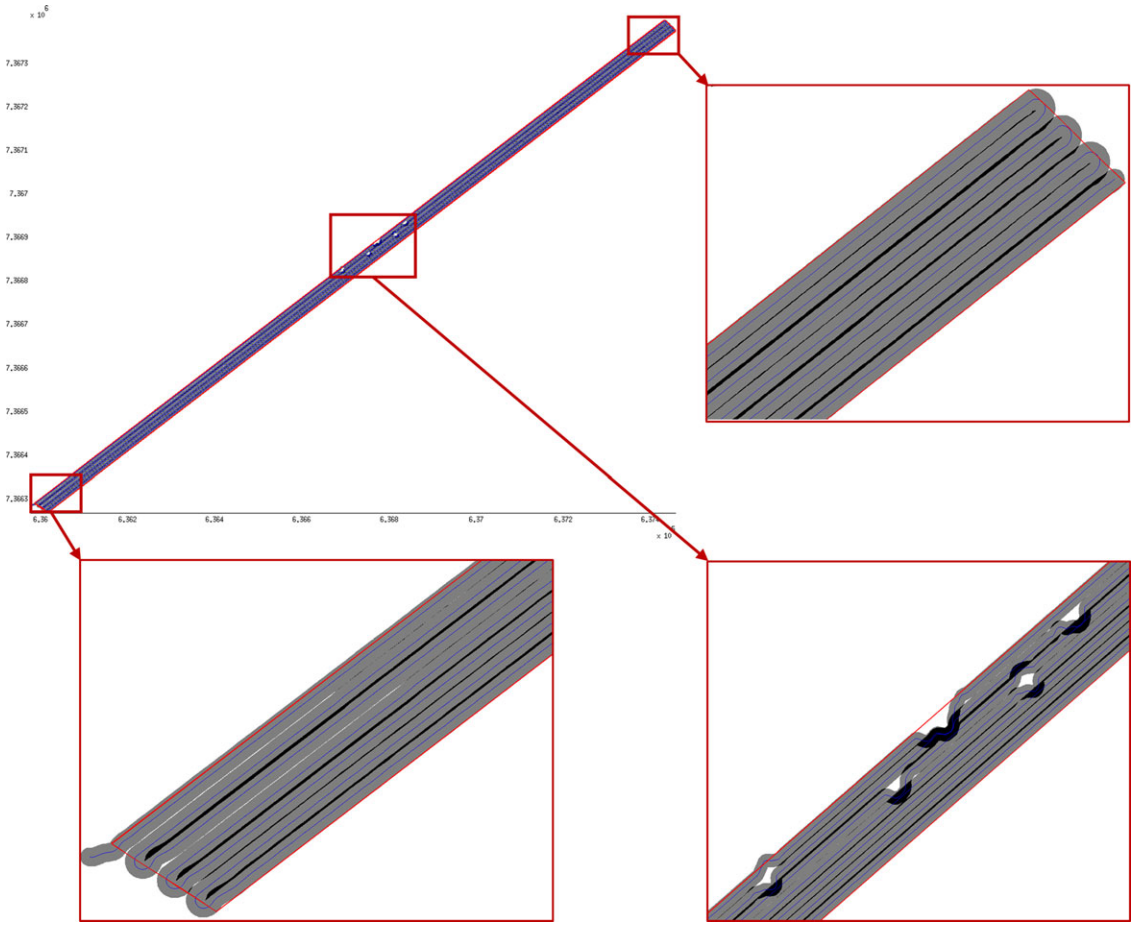

Figure 13. This plot shows the real robot's coverage (gray) with its partition (red) including areas that would be overlapped (black) and missed (white) with spray. The insets show close-up views of the row ends and more clearly show the coverage performance. The middle part shows where the robot avoided several obstacles.

Table III. Table III. Prediction and measurement noise figures used by the particle filter in these experiments.

	Gaussian Error Std. Dev
GPS Position	0.2 m
Odometry Linear Velocity	0.1 m/s
Odometry Angular Velocity	0.04 rad/s
Visual Row Tracking Offset Velocity	0.1 m/s

these blind spots became significant, and they were especially problematic around the utility vehicle/tank obstacle combination. Possible solutions to this problem include the use of multiple stereo camera pairs, or cameras with a wider field of view.

4.1.3. Navigation Results

Figure 19 shows how the obstacle detection system works within the robot's navigation system. On the left side are camera images showing the detection of image novelty. On the right side is a top-down view of the robot's local map showing the trapezoidal vehicle footprint, the obstacles, the shaded row funnel, and the robot's global path. The vehicle is represented as a trapezoid as *move_base* is limited to convex shapes, and we need to account for the trailing boom width. In the first instance, the robot does not detect any obstacles and thus is traveling down the rows within the row funnel. In the second frame, the robot detects the utility vehicle and the lattice planner provides a new global path. In the third frame, the robot has now detected the tank that was

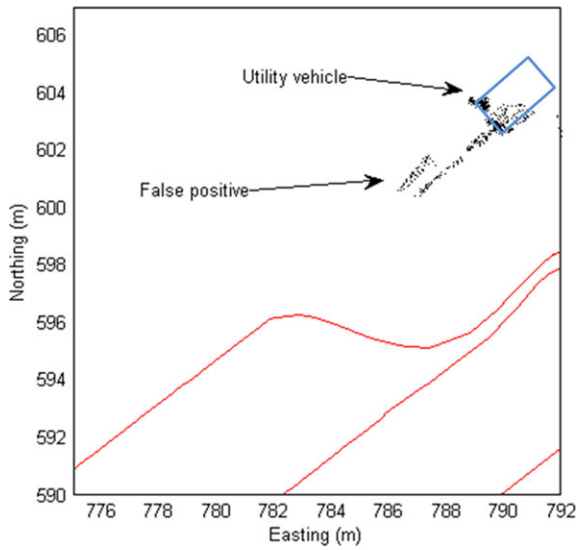

Figure 15. Close-up of the utility vehicle showing the single false positive upon approaching the vehicle. This false positive was due to incorrect matching on the vehicle. The points here were observed from the first pass (lower left of the image).

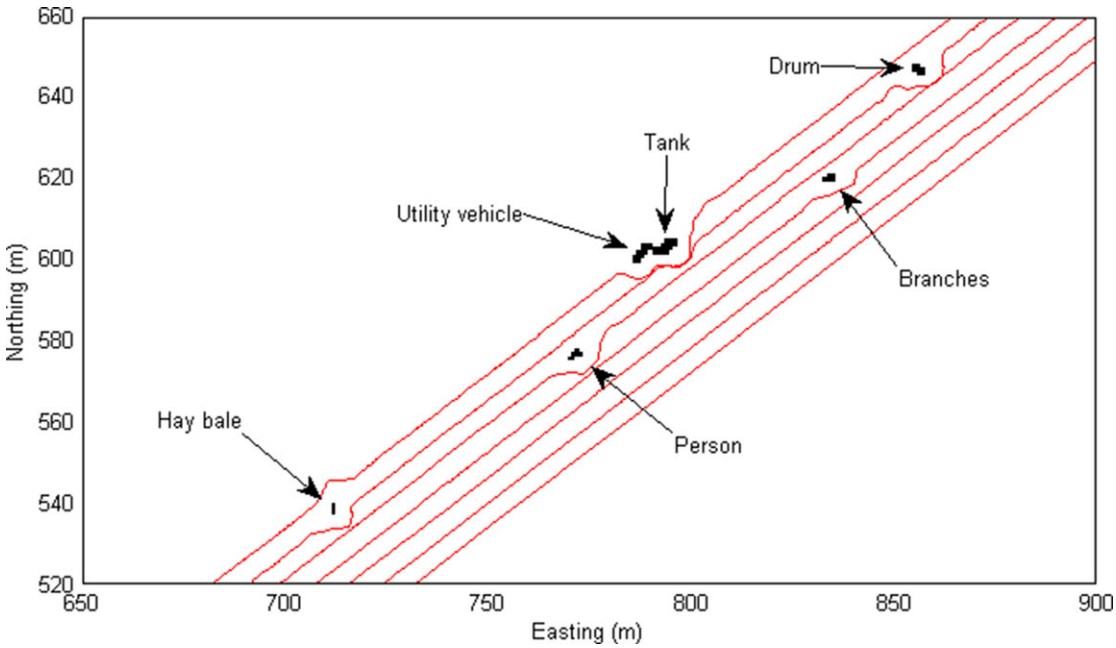

Figure 14. Obstacle map and robot paths. Close-up of the region of the dataset that contained obstacles. The map here is quantized into 1 m blocks, which are marked as containing an obstacle if it was detected at any point in time. For this figure, the robot pose was taken using the ground truth from the Novatel RTK GPS-INS.

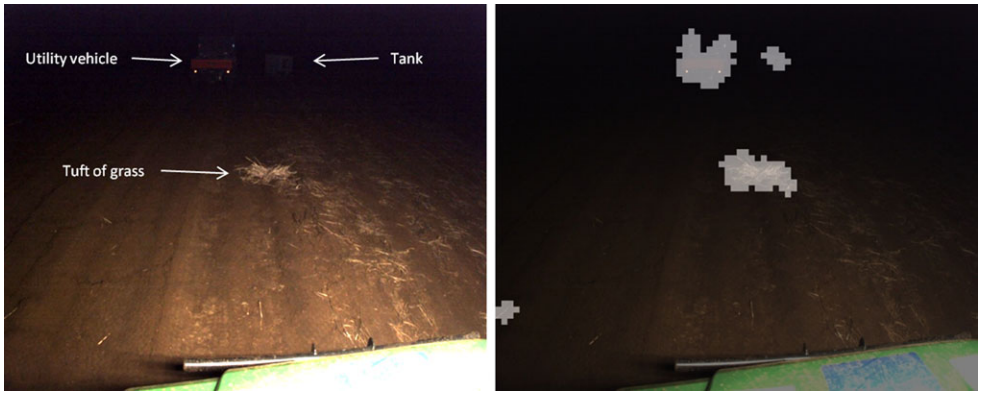

Figure 16. Novelty detection and imagery associated with the false positive. Note that since there is very little texture between the tuft of grass and the vehicle, LIBELAS interpolates these to generate disparities in the middle region, resulting in long false positives extending from the vehicle to the tuft of grass. Only the novel region in the bottom-left is incorrect. Note that the left images have been manually brightened for the reader.

Figure 17. The appearance of the hay bale is close to that of the crop, thus its lower regions, which are in close proximity to the crop, are not detected as novel. The top is successfully detected because it varies more significantly from the local image appearance. This validates the use of the image space positional prior for novelty detection.

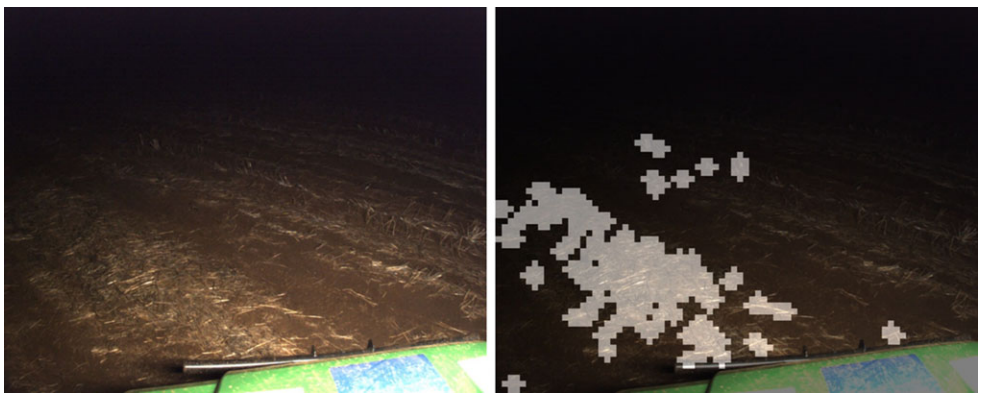

Figure 18. After turning off the straight path to avoid obstacles (not shown in this picture) or turning at the end of the row, the appearance of the environment has locally changed in image space. This leads to higher rates of novelty detection until the model updates to account for these changes, which directly results in a higher chance of false positives. When the appearance is highly dynamic during the actual turn, higher novelty is almost always observed irrespective of learning time because these changes are visible for such a short time compared to the overall experiment duration.

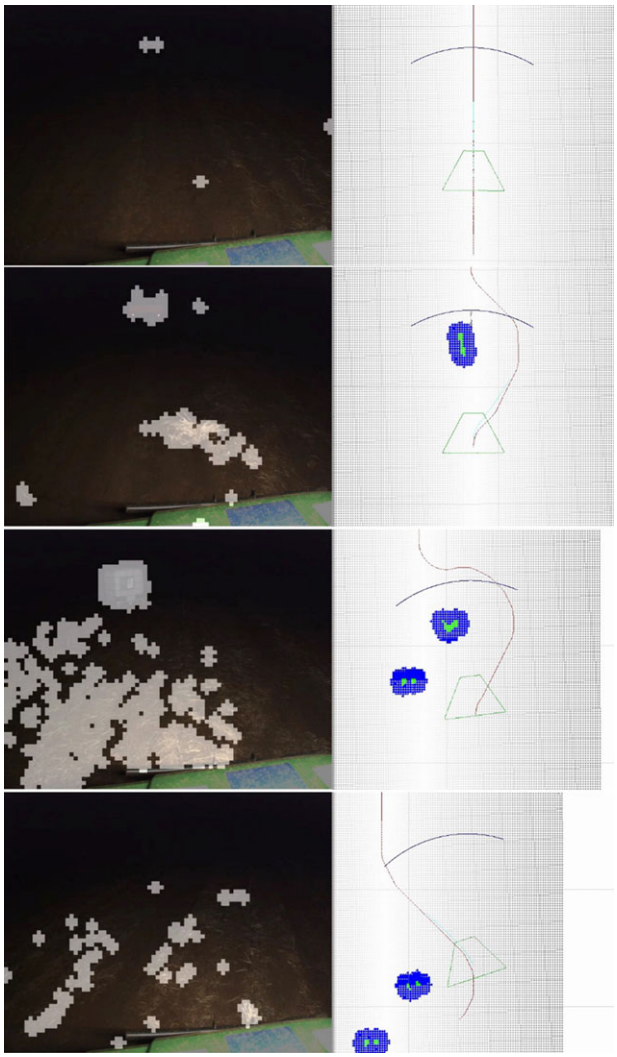

Figure 19. On the left is the camera image with overlaid novelty-base obstacle detection, and on the right is the overhead view of the *move_base* navigation system as given by screenshots from the ROS visualization tool *rviz*. From top to bottom are four instances of time as the robot successfully avoids two obstacles, one of which is originally hidden behind the other. The obstacles are shown in green and the costmap inflation is shown in blue. The green trapezoid outline shows the robot's footprint as used by *move_base*. While the robot is actually a rectangle with a long boom at the rear, representing the robot as a trapezoid has not presented any problems with regard to avoiding obstacles.

initially hidden behind the utility vehicle, and it replans the global path again. In the fourth frame, the robot has moved beyond the obstacles, and the row funnel is bringing the robot quickly back onto the original and correct crop row.

Figure 20. Image from the robot's camera shows the field used to test the visually aided row-following system. The rows are indistinct, and hence they are difficult to detect and track.

4.2. Crop Row Guidance Experiment

This experiment tests the performance of the localization system when using the visual crop row tracking system. In this experiment, we cause artificial GPS outages by removing the GPS input to the particle filter. In each test, the robot is commanded to drive a path parallel with the crop rows at a velocity of approximately 1.4 m/s (5 km/h). The particle filter is allowed to operate normally with GPS input for 250 s, followed by a (software-induced) GPS outage for a period of 300 s.

The two test configurations performed were as follows:

- Test 1. Particle filter relying on dead reckoning using wheel velocity and IMU during GPS outages.
- Test 2. Particle filter relying on visual row tracking during GPS outages.

Each test configuration was performed on three different runs parallel to the crop rows. An example camera image from the experiment is shown in Figure 20, which shows that the crop rows are visually indistinct, making crop row tracking challenging.

The results shown in Figure 21 illustrate the position error of the output of the particle filter (compared to the Novatel ground truth GNSS/INS). Table IV shows the RMS position error for each run. In Test 1, in which the particle filter position is estimated via dead reckoning during GPS outages, the position error grows quickly from several centimeters to several meters. In two of the runs (runs 1 and 3), the test was aborted early due to errors growing so large that the robot left the cropping field and drove toward a fence. These dead reckoning errors are mainly

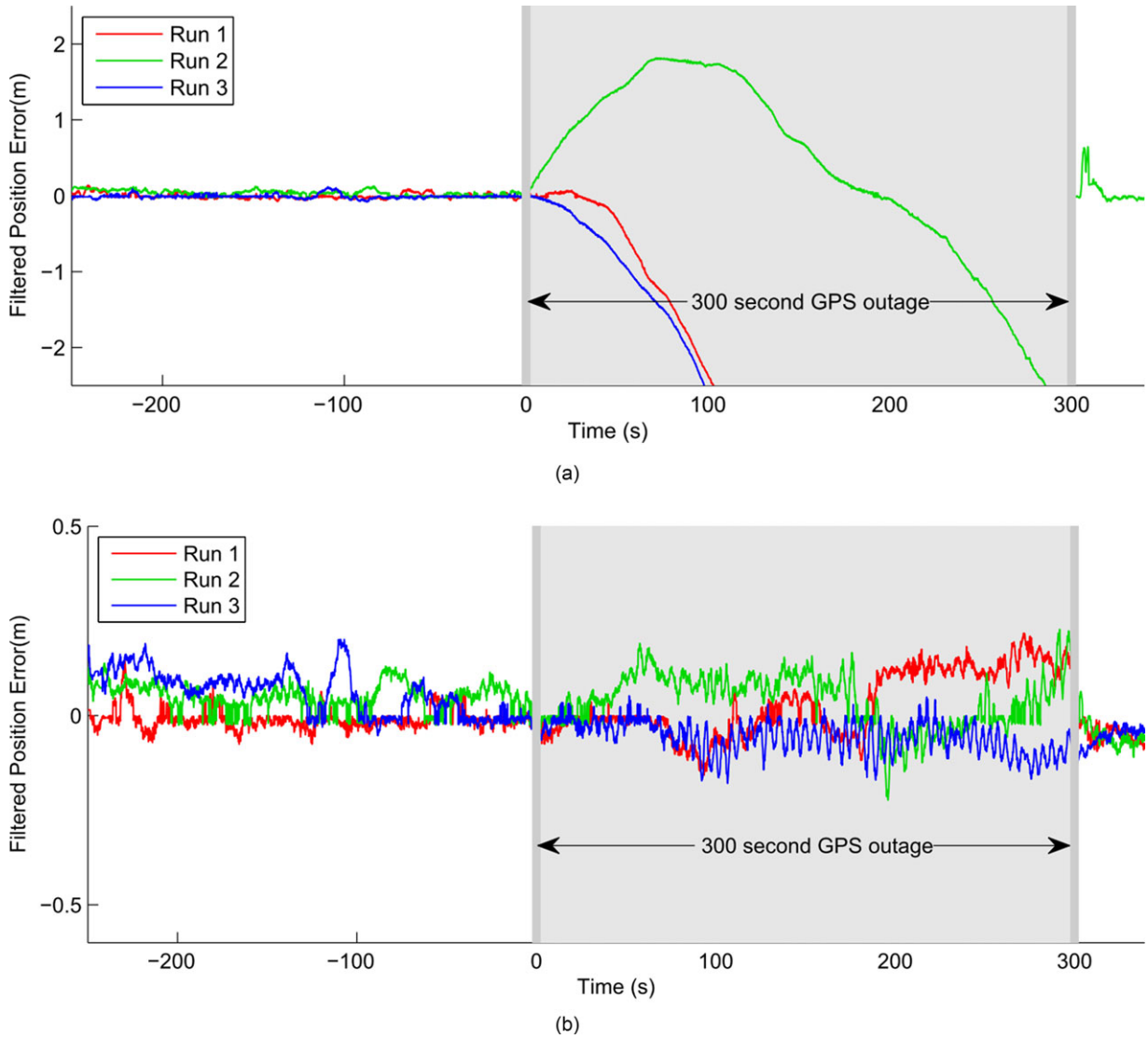

Figure 21. Graph of error in position from the localization system during GPS outages. Visual crop row tracking clearly restricts the growth in position error during GPS outages.

attributed to errors in the heading due to the low-cost IMU.

In Test 2, in which visual crop row tracking was used during GPS outages, the particle filter error remained reasonably constant, with an RMS error of less than 0.1 m. The visual crop row tracking was successfully able to track crop rows for the entire duration of the GPS outage, despite the rows being visually difficult to distinguish. These results clearly demonstrate the usefulness of crop row tracking in reducing position error growth during GPS outages.

5. DISCUSSION

Robotics has the potential to make a meaningful impact in agriculture. This paper presents key technologies for moving beyond the automation of existing farm machinery toward coordinated teams of small, lightweight, inexpensive robots that can manage the fields over the diurnal cycle. The overall goal is not new, and the principles behind it have been described previously (Blackmore et al., 2001). However, in this paper we have demonstrated a computer vision approach using inexpensive sensors to achieve this goal. By lowering the cost of the robot platform, we have brought

Table IV. Filter position error both before and after a 300 s GPS outage. Visual crop row track dramatically reduces position error during GPS outages.

Filter Position Error (m)	Run	Before GPS Outage	During GPS Outage
Test 1 – Dead reckoning during GPS outage	1	0.0548	5.43
	2	0.0535	4.46
	3	0.0304	7.94
Test 2 – Crop row tracking during GPS outage	1	0.0286	0.0891
	2	0.0647	0.0860
	3	0.0858	0.0747

the overal goal closer to a commercial reality. The approach has been well tested in numerous coverage tasks across several weeks of day and night field trials. This paper reports on one particular coverage experiment conducted at night and lasting approximately three hours. Two innovations are presented: visual novelty-based obstacle avoidance and visually aided localization.

The results demonstrate that novelty-based obstacle detection using stereo vision is appropriate for detecting obstacles in cropping fields as the appearance of cropping fields is relatively uniform compared to the appearance of typical obstacles. Since the novelty detector's model of the environment is continuously updated, it generalizes well across different types of crop fields and lighting conditions. A caveat is that the obstacles need to differ in appearance; however, many obstacles such as hay bales are visually similar to the crop. Future work will investigate using a wider range of cues, such as structure, in the novelty descriptor.

In this particular experiment, the obstacle detector and path planner were tested only on static obstacles. As the detector is stateless and uses no information about the robot's egomotion, there will be no difference between dynamic obstacles and static obstacles from the obstacle detector's perspective. However, this means that the path planner can only account for the current state of the obstacles and would be unable to predict intersecting paths and avoid future collisions. In practise, due to the robot's slow speed, this has not been an issue. Of more importance is the detection range, the processing rate, and the field of view.

Computing time was a limiting factor for the obstacle detection system. Despite the significant speedups achieved in stereo matching via the use of selective focus, the number of separate processes running due to the ROS achitecture used most of the available CPU time. In particular, when passing images between a large number of ROS nodes, a significant amount of computing time was spent on background memory copying. This was solved by integrating several nodes together to minimize the amount of redundant copying. This allowed the obstacle detection system to achieve a 10 Hz frame rate with standard computer hardware.

The obstacle detector functioned well when it was sufficiently tuned. However, parameter tuning was a time consuming process to remove false positives. Some of this was due to the large amounts of imaging noise present, as nighttime operation required the gain of the cameras to be set at maximum. There was also a degree of interframe inconsistency due to the irregular operation of the pulsed light source.

The localization system was able to guide the robot to perform a coverage task with sufficient accuracy despite using only inexpensive hardware. The visually aided localization system was able to successfully guide a robot along crop rows without the need for a GPS-INS. Because the technique deliberately avoids segmenting crop rows and performs only frame-to-frame tracking, it is applicable to a wide variety of crop types. However, these design decisions that allow reliable operation in difficult conditions also cause several limitations, namely that the crop offset estimation drifts slowly over time, and that the method is unable to prevent the vehicle from driving on the crop rows. These tradoffs were acceptable for the environment and experiments performed in this paper. However, future work will look into extending the current visual row tracking algorithm to address these limitations while still mainitaining the ability to track crop rows in visually difficult conditions.

While the visual navigation technique presented in this paper is designed for fields with crop or stubble rows, it may be applicable in other tasks in which tracking parallel texture is useful. For example, we inadvertently discovered while traveling between fields that the visual crop row tracking method can successfully track wheel tracks and ruts on a variety of unpaved roads.

An alternative method for visual navigation during GPS outages would be to use stereo visual odometry (Kuemmerle, Grisetti, Strasdat, Konolige, & Burgard, 2011), which has achieved impressive results, with errors often less than 0.1% of distance traveled. However, one disadvantage of using visual odometry instead of visual crop row tracking in our application is that a small error in the particle filter's heading estimate at the moment when the GPS measurements became unavailable would result in driving at an angle from the intended crop row.

The robot platform, an existing electric vehicle modified for autonomous use, is unsuited mechanically for this field environment due to issues such as ground clearance and wheel spacing. However, it has successfully allowed us to focus on the development of appropriate algorithms. Australian farm environments are harsh on vehicles and robots due, in particular, to high temperatures, vibration, and dust. A substantial part of the field test time was spent adding additional cooling to the vehicle's controller, motor, and computer ebox.

Two factors effected our ability to run long-term experiments. Vehicle tires operating on a farm environment have a high rolling resistance of approximately 0.15–0.2, which is approximately 15 times greater than the rolling resistance of tires on pavement. This substantially reduced battery life to 3–4 h of continuous operation. As the vehicle uses lead acid batteries, the recharge time was very high, approximately 12–14 h. This had an effect on our ability to run a single nighttime long-term continuous experiment, as a failure half-way through meant waiting another day.

The second factor is an early decision made in the design of the robot's navigation system. For safety reasons, we decided to prevent the robot from being able to reverse. This is because there are no sensors for detecting obstacles that are behind the robot, and we wanted the area behind the robot to be a safe place for humans. While the robot has never collided with an obstacle, there have been numerous times, particularly at night, when the obstacle was detected too late for the robot to be able generate a collision-free plan given the Ackerman kinematic constraints. This meant that human assistance was needed, and for this paper it entailed restarting the experiment. However, other robot drive system configurations, such as differential drive, would be able to more easily handle these situations.

The paper also presents how to modify the ROS *move_base* architecture and navigation stack to support an Ackermann robot that works outdoors in a field environment. While the ROS navigation stack provided a flexible navigation system from which to develop, it is primarily designed for navigating differential drive or omnidirectional vehicles in an indoor setting using a laser scanner. To make this system suitable for our agricultural vehicle, we made modifications to nearly all the ROS navigation stack subcomponents, including replacing the global and local planners, modifying the costmap to include potential fields for row following, replacing the existing particle filter with our own particle filter (which uses GPS), and adjusting the pose transforms.

It is arguable whether a robot used for agriculture requires path planning for obstacle avoidance within the relative complex ROS *move_base* architecture, rather than just detecting obstacles and waiting for human intervention. The disadvantage of driving around obstacles is that the robot drives across rows, causing damage to the crop. The advantage is that the robot can continue autonomously without human interaction, which is important for a large number of robots. Also, in the future we plan to incorporate docking and refilling, which will bring the robot closer to obstacles.

While time synchronization between sensors was adequate for this experiment, more accurately synchronized sensor measurements would allow closer coupling of sensor data. ROS systems are by design distributed across multiple processes (and potentially computers) with variable network delays, and they use non-real-time operating systems. These characteristics make accurate time synchronization between sensors difficult without specialized hardware. In future work, we intend to examine methods that can be used to provide better synchronization between the various sensors in our system.

ACKNOWLEDGMENTS

This work was supported in part by the Australian Research Council Linkage Project LP110200375 "Robotics for Zero-Tillage Agriculture" awarded to Queensland University of Technology, SwarmFarm Robotics, and The University of Sydney. The authors would like to acknowledge Kyran Findlater for designing and building the strobe light system, as well as RoPro Design for preparing the vehicle for autonomous operation.

REFERENCES

Angelova, A., Matthies, L., Helmick, D., & Perona, P. (2007). Learning and prediction of slip from visual information. Journal of Field Robotics, 24(3), 205–231. doi: 10.1002/rob.20179

Åstrand, B., & Baerveldt, A.-J. (2005). A vision based row-following system for agricultural field machinery. Mechatronics, 15, 251–269. doi: 10.1016/j.mechatronics.2004.05.005

Bakker, T., Wouters, H., van Asselt, K., Bontsema, J., Tang, L., Müller, J., van Straten, G., & Asselt, K. V. (2008). A vision based row detection system for sugar beet. Computers and Electronics in Agriculture, 60, 87–95. doi: 10.1016/j.compag.2007.07.006

Ball, D., Ross, P., English, A., Patten, T., Upcroft, B., Fitch, R., Sukkarieh, S., Wyeth, G., & Corke, P. (2013). Robotics for sustainable broad-acre agriculture. Paper presented at the 9th Conference on Field and Service Robotics, Brisbane, Australia.

Bell, T. (2000). Automatic tractor guidance using carrier-phase differential GPS. Computers and Electronics in Agriculture, 25, 53–66.

Bossu, J., Gée, C., Jones, G., & Truchetet, F. (2009). Wavelet transform to discriminate between crop and weed in perspective agronomic images. Computers and Electronics in Agriculture, 65(1), 133–143.

Biber, P., Weiss, U., Dorna, M., & Albert, A. (2012, October). Navigation system of the autonomous agricultural robot Bonirob. In Workshop on Agricultural Robotics: Enabling

Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal.

Billingsley, J., & Schoenfisch, M. (1997). The successful development of a vision guidance system for agriculture. Computers and Electronics in Agriculture, 16, 147–163.

Blackmore, S., Have, H., & Fountas, S. (2001). A specification of behavioural requirements for an autonomous tractor. Paper presented at the 6th International Symposium on Fruit, Nut and Vegetable Production Engineering Conference, Potsdam, Bornim, Germany.

Broggi, A., Cappalunga, A., Caraffi, C., Cattani, S., Ghidoni, S., Grisleri, P., Porta, P. P., Posterli, M., & Zani, P. (2010). Terramax vision at the urban challenge 2007. IEEE Transactions on Intelligent Transportation Systems, 11(1), 194–205.

Chateau, T., Debain, C., Collange, F., Trassoudaine, L., & Alizon, J. (2000). Automatic guidance of agricultural vehicles using a laser sensor. Computers and Electronics in Agriculture, 28(3), 243–257.

Choset, H. (2001). Coverage for robotics—A survey of recent results. Annals of Mathematics and Artificial Intelligence, 31(1–4), 113–126.

Cohen, B. J., Chitta, S., & Likhachev, M. (2010). Search-based planning for manipulation with motion primitives. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK.

Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., & Bradski, G. (2006). Self-supervised monocular road detection in desert terrain. Paper presented at the Robotics: Science and Systems II, Philadelphia.

English, A., Ball, D., Ross, P., Upcroft, B., Wyeth, G., & Corke, P. (2013). Low cost localisation for agricultural robotics. Paper presented at the Australasian Conference on Robotics and Automation (ACRA), Sydney.

English, A., Ross, P., Ball, D., & Corke, P. (2014). Vision based guidance for robot navigation in agriculture. Paper presented at the International Conference on Robotics and Automation (ICRA), Hong Kong.

Filitchkin, P., & Byl, K. (2012). Feature-based terrain classification for LittleDog. Paper presented at the IEEE International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal.

Fleischmann, P., Föhst, T., & Berns, K. (2013). Detection of field structures for agricultural vehicle guidance. KI-Künstliche Intelligenz, 27(4), 351–357.

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. Robotics and Autonomous Systems, 61(12), 1258–1276.

Geiger, A., Roser, M., & Urtasun, R. (2011). Efficient large-scale stereo matching. Paper presented at the Computer Vision-ACCV 2010.

Gerrish, J. B., Stockman, G. C., Mann, L., & Hu, G. (1986). Path-finding by image processing in agricultural field operations (no. 861455). SAE Technical Paper.

Hadsell, R., Sermanet, P., Ben, J., Erkan, A., Scoffier, M., Kavukcuoglu, K., Muller, U., & LeCun, Y. (2009). Learning long-range vision for autonomous off-road driving. Journal of Field Robotics, 26(2), 120–144.

Hague, T., & Tillett, N. D. (2001). A bandpass filter-based approach to crop row location and tracking. Mechatronics, 11, 1–12.

Han, S., Zhang, Q., Ni, B., & Reid, J. F. (2004). A guidance directrix approach to vision-based vehicle guidance systems. Computers and Electronics in Agriculture, 43, 179–195.

Hanawa, K., Yamashita, T., Matsuo, Y., & Hamada, Y. (2012). Development of a Stereo Vision System to Assist the Operation of Agricultural Tractors. Japan Agricultural Research Quarterly: JARQ, 46(4), 287–293.

Hiremath, S., Evert, F. V., Heijden, G., Braak, C. J. F. T., & Stein, A. (2012). Image-based particle filtering for robot navigation in a maize field. Paper presented at the Workshop on Agricultural Robotics, IEEE International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal.

Hiremath, S., van Evert, F. K., ter Braak, C., Stein, A., & van der Heijden, G. (2014). Image-based particle filtering for navigation in a semi-structured agricultural environment. Biosystems Engineering, 121, 85–95.

Jensen, K., Larsen, M., Simonsen, T., & Jørgensen, R. (2012). Evaluating the performance of a low-cost GPS in precision agriculture applications. Paper presented at the First International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture (RHEA-2012), Pisa, Italy.

Khan, Y. N., Komma, P., & Zell, A. (2011). High resolution visual terrain classification for outdoor robots. Paper presented at the IEEE International Conference on Computer Vision Workshops (ICCV Workshops).

Kim, D., Sun, J., Oh, S. M., Rehg, J. M., & Bobick, A. F. (2006). Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA).

Kuemmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). g2o: A general framework for graph optimization. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Laws, K. (1980). Rapid texture identification. Paper presented at the Society of Photo-Optical Instrumentation Engineers (SPIE), Image Processing for Missile Guidance.

Libby, J., & Kantor, G. (2011). Deployment of a point and line feature localization system for an outdoor agriculture vehicle. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA), Shanghai.

Marchant, J. A. (1996). Tracking of row structure in three crops using image analysis. Computers and Electronics in Agriculture, 15, 161–179.

Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., & Konolige, K. (2010). The office marathon: Robust navigation in an indoor office environment. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA).

Milella, A., & Reina, G. (2014). 3D reconstruction and classification of natural environments by an autonomous vehicle

using multi-baseline stereo. Intelligent Service Robotics, 7(2), 79–92.

Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. Journal of Field Robotics, 26, 651–668.

Ollis, M., & Stentz, A. (1997). Vision-based perception for an automated harvester. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Olsen, H. J. (1995). Determination of row position in small-grain crops by analysis of video images. Computers and Electronics in Agriculture, 12, 147–162.

Osterman, A., Godeša, T., & Hočevar, M. (2013). Introducing low-cost precision GPS/GNSS to agriculture. Paper presented at the 41st International Symposium on Agricultural Engineering, Opatija, Croatia.

Pedersen, S. M., Fountas, S., Have, H., & Blackmore, B. S. (2006). Agricultural robots—System analysis and economic feasibility. Precision Agriculture, 7, 295–308.

Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K., & Stentz, A. (2002). The demeter system for automated harvesting. Autonomous Robots, 13, 9–20.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source robot operating system. Paper presented at the Workshop on Open Source Software, IEEE International Conference on Robotics and Automation (ICRA).

Ranganathan, A., Matsumoto, S., & Ilstrup, D. (2013). Towards illumination invariance for visual localization. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA).

Reid, J. F., & Searcy, S. W. (1987). Vision-based guidance of an agricultural tractor, IEEE Control Systems Magazine (Vol. 7, pp. 39–43).

Reina, G., & Milella, A. (2012). Towards autonomous agriculture: Automatic ground detection using trinocular stereo vision. Sensors, 12(9), 12405–12423.

Ross, P., English, A., Ball, D., Upcroft, B., Wyeth, G., & Corke, P. (2013). A novel method for analysing lighting variance. Paper presented at the Australian Conference on Robotics and Automation (ACRA), Sydney.

Ross, P., English, A., Ball, D., Upcroft, B., Wyeth, G., & Corke, P. (2014). Novelty-based visual obstacle detection in agriculture. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA).

Rovira, F., Más, Q. Z., & Reid, J. F. (2004). Automated agricultural equipment navigation using stereo disparity images. Transactions of the American Society of Agricultural Engineers (ASAE), 47, 1289–1300.

Rovira-Más, F., Zhang, Q. & Reid, J. F., (2008). Stereo vision three-dimensional terrain maps for precision agriculture. Computers and Electronics in Agriculture, 60(2), 133–143.

Sinden, J., Jones, R., Hester, S., Odom, D., Kalisch, C., James, R., & Cacho, O. (2004). The economic impact of weeds in Australia. Technical Series, 8.

Søgaard, H. T., & Olsen, H. J. (2003). Determination of crop rows by image analysis without segmentation. Computers and Electronics in Agriculture, 38, 141–158.

Stempfhuber, W., & Buchholz, M. (2011). A precise, low-cost RTK GNSS system for UAV applications. Remote Sensing and Spacial Information Sciences, XXXVIII, 14–16.

Stentz, A., Dima, C., Wellington, C., Herman, H., & Stager, D. (2002). A system for semi-autonomous tractor operations. Autonomous Robots, 13, 87–104.

Takai, R., Noguchi, N., Yang, L., & Zhang, Z. (2011). Crawler-type robot tractor using multi-GNSS with QZSS. Paper presented at the International Commission of Agricultural and Biosystems Engineering–Agricultural Engineers (CIGR-AgEng).

Takasu, T., & Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. International Symposium on GPS/GNSS (pp. 4–6). Jeju, Korea: International Convention Centre.

Taylor, J. (1983). Benefits of permanent traffic lanes in a controlled traffic crop production system. Soil and Tillage Research, 3, 385–395.

Tillett, N. D., Hague, T., & Miles, S. J. (2002). Inter-row vision guidance for mechanical weed control in sugar beet. Computers and Electronics in Agriculture, 33, 163–177.

Torii, T. (2000). Research in autonomous agriculture vehicles in Japan. Computers and Electronics in Agriculture, 25, 133–153.

Valgren, C., & Lilienthal, A. (2007). SIFT, SURF and seasons: Long-term outdoor localization using local features. Paper presented at the Mobile Eobots (ECMR), European Conference.

Vernaza, P., Taskar, B., & Lee, D. D. (2008). Online, self-supervised terrain classification via discriminatively trained submodular Markov random fields. Paper presented at the IEEE International Conference on Robotics and Automation (ICRA) Pasadena, CA.

Warren, M., McKinnon, D., & Upcroft, B. (2013). Online calibration of stereo rigs for long-term autonomy. Paper presented at the International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany.

Weiss, U., & Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. Robotics and Autonomous Systems, 59(5), 265–273.

Wellington, C., Courville, A., & Stentz, A. (2005). Interacting Markov random fields for simultaneous terrain modeling and obstacle detection. Paper presented at the Robotics: Science and Systems, Cambridge, MA.

Zhang, J., Kantor, G. A., Bergerman, M., & Singh, S. (2012). Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal.