Mapping Orchards for Autonomous Navigation

Ji Zhang, Silvio Maeta, Marcel Bergerman, and Sanjiv Singh

Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA

Written for presentation at the

2014 ASABE and CSBE/SCGAB Annual International Meeting

Sponsored by ASABE

Montreal, Quebec Canada

July 13 – 16, 2014

Abstract. *Autonomous orchard vehicles have been shown to increase worker efficiency in tasks including pruning, thinning, tree maintenance, and pheromone placing. To cover entire blocks, they must be able to repeatedly exit an orchard row, turn, and enter the next. The authors' experience deploying autonomous vehicles for five years in commercial and research orchards shows that, when the map of the block is available, navigation errors decrease and the vehicle's probability of finding the next row increases. From a cost perspective, it is important that the map be built without expensive surveying equipment, and preferably using the same sensors used for vehicle guidance. Here, we present a landmark-based method that creates a local map of the block in which the vehicle operates using the laser scanner already present on the vehicle. We describe the mapping procedure and present results obtained in commercial orchards.*

Keywords. *Agricultural Robotics; Autonomous Orchard Vehicle; Autonomous Navigation; Mapping; Lidar.*

Introduction

Fruit production in the US has faced increasing pressure from international competitors, at the same time when labor availability from season to season has become uncertain. In the past decade, engineers and extension educators have experimented with autonomous orchards vehicles as a means to increase farm worker efficiency, thus reducing labor costs and smoothing out workforce requirements throughout the year.

From a very low-level robotics perspective, such vehicles must be able to follow an orchard row, exit it, turn, and enter the next row, repeatedly and reliably, all day long. There are several examples of autonomous row following in orchards and similar agricultural settings in the scientific literature—e.g., Hiremath et al., 2012, Bakker et al., 2008, Astrand & Baerveldt, 2005, Satow et al., 2004, Biber et al., 2012, and Morehead et al., 2012. In our previous work, an accurate tree row detection and following algorithm has been developed (Zhang et al., 2003). The algorithm extracts the structure of the tree rows on both sides of the vehicle, and commands the vehicle to follow the rows either on the center or keeping a constant distance to one side. Experiments have shown accuracy of a few centimeters in terms of lateral offset errors.

Experience gathered over five years deploying autonomous orchard vehicles in research and commercial orchards in Washington and Pennsylvania tell us that detecting the end of the row, turning, and entering the

next row are the hardest aspects of block coverage if the vehicle is to operate in a variety of crops and canopies. Mapping the block's local metric structure (i.e., where each tree row begins and their length) has shown to improve significantly our vehicles' ability to traverse entire blocks for hours on end.

This paper presents a method to create such local map for autonomous orchard navigation. It relies on the placement of strips of retro-reflective tape on the posts at the end of the rows, which are easily detected by the lidar on the vehicles. These artificial landmarks indicate to the vehicle where each row ends and the next one begins, thus increasing its probability of finding the next row and entering it successfully. Differently from other works (e.g., Libby and Kantor, 2011), the mapping method presented here does not require GPS or the use of expensive surveying equipment. Instead, it relies only on the sensors already installed on the vehicle for row following purposes, i.e., a lidar and wheel encoders. The mapping method takes as input readings from the lidar and encoders as the vehicle is manually driven around the block for at least two full rounds. The data collected are processed off-line to generate the map, which is then permanently stored on the vehicle's onboard computer.

The mapping method has been tested in different commercial orchards, with conditions varying from mostly level to highly sloped terrain. The results indicate that accurate maps can be reliably created. Furthermore, we show in this paper that the localization errors using those maps are within 50 cm of the ground truth provided by a highly accurate GPS-assisted inertial navigation system.

Vehicle, Sensors, and Landmarks

The autonomous orchard vehicle used in this study is shown in Figure 1 (left). It is based on a Toro Workman MDE chassis retrofitted as a drive-by-wire platform. It is equipped with steering and driving wheel encoders that provide odometry measurements. A proportional-integral controller is used in the low-level steering and speed control loops. The trajectory control is implemented by the vehicle tracking a look-ahead point 1.5 m ahead of it, a technique known as pure pursuit (Roth & Batavia, 2002).

A custom-made 3D lidar (Figure 1, center) is mounted on the front of the vehicle, pointing forward. The lidar consists of a Hokuyo laser scanner connected to a motor. The laser scanner has a field of view of 180° with 0.25° resolution and 40 Hz scan rate. An embedded microcontroller controls the rotation of the motor at an angular speed of 100°/s and measures the rotation angle through an optical quadrature encoder with a resolution of 0.25°. The microcontroller also manages the time synchronization between the laser scanner and the optical encoder. The laser scanner provides a synchronization pulse that indicates the start of each scanning cycle. The synchronization pulse triggers the microcontroller to report the optical encoder reading. Through a serial cable, the data is sent to a computer, where the laser points are projected into 3D space. When projecting the laser points, the rotation angle of the laser scanner is found by linearly interpolating the encoder readings within the laser scanning cycles.

To produce accurate orchard maps, we install retro-reflective landmarks on the posts at the end of each tree row (Figure1, right), at the height of the laser scanner (about 50 cm from the ground). In orchards where there are no end posts, the landmarks are attached to a PVC pipe. In the lidar point cloud, points returned from the landmarks have a high intensity and stand out distinctively from all other points, such as trees and ground. This helps us identify the landmarks easily from the lidar point cloud.
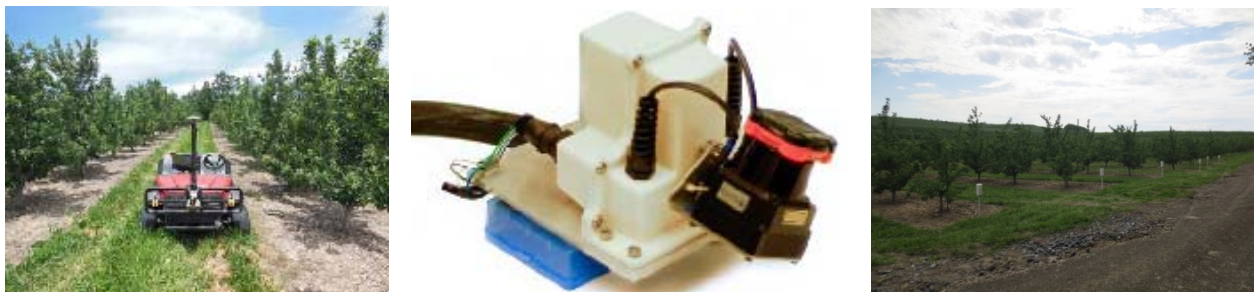

Figure 1. Left: Autonomous orchard vehicle. Center: 3D lidar sensor. Right: Landmarks installed at the ends of the tree rows.

Autonomous Navigation

Tree Row Detection and Following

To navigate successfully in orchards, the vehicle must detect the tree rows from the point cloud acquired by the lidar. For tasks such as autonomous mowing, spraying, and fruit bin transporting, the vehicle must follow a line parallel to the centerline between two tree rows, possibly with an offset toward one side. For tasks such as pruning, thinning, and others performed by workers riding on the vehicle, it must be able to keep a fixed distance to the canopy. The technology necessary to accomplish these two driving modes is described in our previous work (Zhang et al., 2003). For the sake of completeness, it is briefly summarized here.

The first driving mode requires detection of the tree rows on both sides of the vehicles and their parameterization as two parallel lines. Our method is based on running the RANSAC (random sample consensus) algorithm (Fischler & Bolles, 1981) on the point clouds on the right and the left of the lidar (Figure 2, left). At each iteration of the algorithm, three points are randomly selected from the point cloud, two from one side and one from the other. Two parallel lines are extracted from these three points. The distances from all other points in the point cloud to those two lines are calculated, and a subset of the points is selected as inliers. Three new points are randomly selected, and the process repeats until it converges or a pre-defined number of iterations is reached. The vehicle is then commanded to follow the centerline between the final two lines (represented in red in Figure 2, right), plus an offset to one side or another depending on the task.

The second driving mode uses direct tree trunk or canopy detection. Here we use a particle filter to lump the point cloud information into discrete trees. The particle filter uses vehicle odometry as the motion model and the lidar point cloud as the observation model. The distribution of the points on a tree trunk and canopy is modeled as a Gaussian distribution. The particles are resampled iteratively and gradually converge to the location of each tree (Figure 2, right). Note that, as the vehicle drives, the tree trunks closest to the vehicle "jumps" from one tree to the next one in the row. This is represented in Figure 2 (right) by the black circles "jumping" to the white ones. The vehicle is then commanded to keep a fixed distance from the trunks (or canopy).
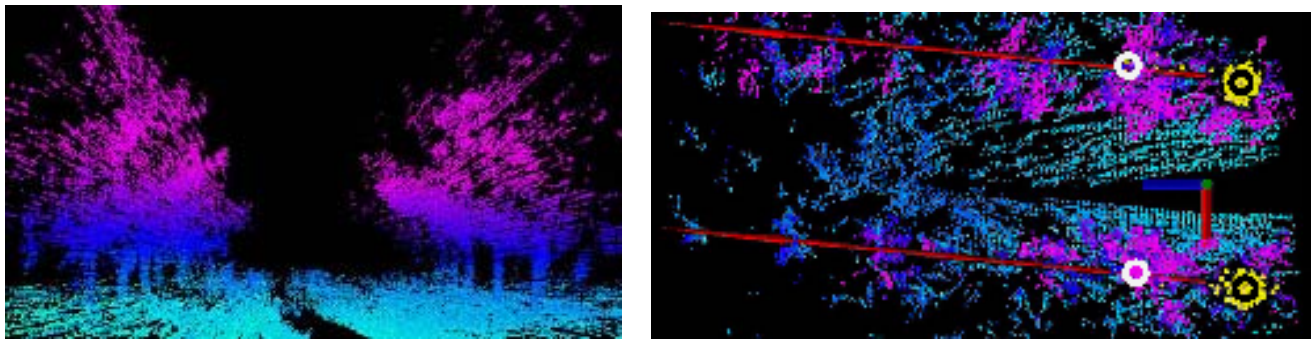

Figure 2. Left: Lidar point cloud acquired from orchard tree rows. Right: Fitted straight lines representing tree rows (red lines) and particles surrounding the closest tree trunks on each side of the vehicle (yellow dots). The white and black circles are manually-labeled tree trunks. As the vehicle drives forward, the particles switch from the black circles to the white circles.

Path Generation and Turning

In order to evaluate the result from the proposed mapping method in a real navigation task, a series of autonomous block traversal tests were performed in commercial orchards. In this subsection we describe the path generation module used in those tests, which simply aims to take the vehicle from one row to the next.

The path planning for block traversal can be decomposed in two types of segments that are concatenated to generate the full trajectory: row following and end-of-row turning trajectory segments (Figure 3). The row following trajectory is a straight line between the two trees rows being traversed. It is possible to set a lateral offset to make the autonomous vehicle perform row following closer to one of the tree rows.
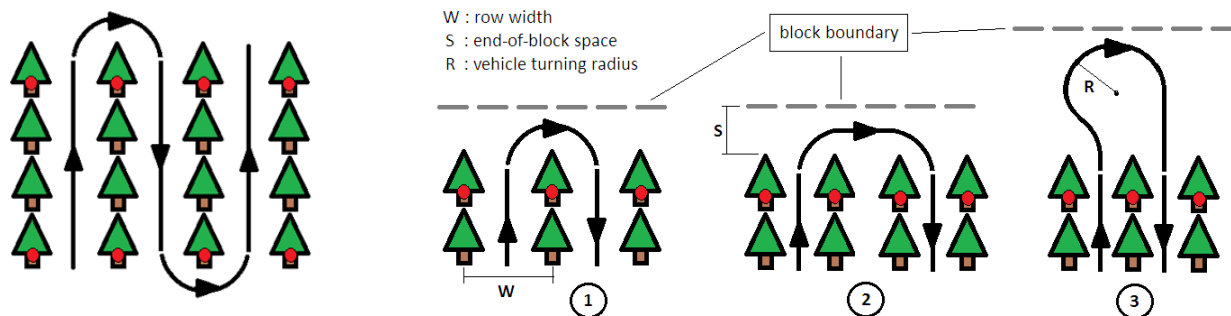
Block traversal trajectory segments.   *End-of-row turning strategies.*

Figure 3. Left: Block traversal trajectories are composed of row following and turning segments. Right: Turning segments can be (1) simple U-turns, (2) skip-rows turns, or (3) asymmetric "bulb" turn. Red circles represent the reflective landmarks.

Several end-of-row turning trajectories were implemented to handle spatial constraints present in real orchards: row width (W) and headland length (S). If the row width is less than the vehicle turning radius (R), the vehicle can't exit one row and enter the next row executing a simple U-turn. Likewise, if the headland length is less than the vehicle's turning radius, then the vehicle can't execute the turn after exiting the row completely, unless it backs up during the process. Table 1 summarizes the different types of turning strategies implemented to account for variations in row width and headland length.

Table 1. Three turning strategies used to deal with variations in orchard row width and headland length.

Row Width	Headland Length	Turning profile	Remarks
W > R	Any (*)	Simple U-turn	Vehicle can exit current row and enter adjacent row. (*) Must be long enough to allow safe vehicle traversal.
W < R	S < R	Skip-rows turn	A simple U-turn can't be used due to the restricted space, therefore it is necessary to skip one or more rows to execute a turn.
W < R	at least 3 x R	Asymmetric bulb turn	This turn allows the vehicle to enter the adjacent row but requires a large space at the headland.

Turns that include back up maneuvers were not considered because the only sensor that scans the environment is mounted pointing forward. Allowing the vehicle to move backwards to perform a "blind" turn can be dangerous, especially in the presence of moving obstacles such as persons or machinery.

The turning strategy defines how the vehicle will exit the current row and enter the next row. Since the reflective landmarks are positioned at the end of the rows, the vehicle corrects its position and orientation when it is exiting or entering a row. In the current implementation, it is necessary that the lidar detects at least two reflective landmarks to perform position correction. During experimental tests, the best correction results were achieved when executing turns in which the vehicle exits the row completely, executes the turn, and aligns itself with the next row before entering. This allows the sensor to capture both reflective landmarks when exiting the current row and also when entering the next one. This is only possible, however, when the headland is long enough. When the vehicle executes a turn in very limited spaces, sometimes just one of the landmarks is visible, meaning that the position estimation is not corrected by the localization system.

Orchard Block Mapping

This section presents the main contribution of this paper, i.e., an orchard block mapping method that is used to enable repeatable and reliable row-to-row navigation. In a nutshell, the method consists of (1) registering the point cloud of the canopy using encoder odometry, (2) detecting and mapping the landmarks, and (3) correcting the relative position of the landmarks via loop closure. An additional step is performed when wheel slip is significant, for example when the terrain is very steep. These steps are described in what follows.

Encoder Odometry

The orchard vehicle used in this work (Figure 1) is equipped with a steering wheel encoder, to measure the steering angle, and a driving wheel encoder, to measure the angular velocity of the back wheels. By combining the readings from the two encoders we compute the velocity of the vehicle as follows. Let v be the vehicle

speed measured by the driving wheel encoder, and let θ be the current heading of the vehicle. The vehicle's linear velocity can be calculated as

$$v_x = v \cos \theta$$

$$v_y = v \sin \theta$$

where v_x and v_y are the velocities in the x- and y- directions, respectively, with respect to the ground. Let α be the turning angle of the steering wheel; the vehicle's angular velocity can be expressed as

$$\dot{\theta} = \frac{\alpha v}{L}$$

where L is the distance between the front and back wheels' axles. Integrating v_x, v_y, and θ over time, a process known as encoder odometry, yields the vehicle's new position and orientation.

Note that the encoder odometry only measures the velocity on the ground plane, i.e. the planar motion of the vehicle assuming the ground is flat. Considering the fact that orchard blocks are mostly flat within a short range, this limitation has not been an issue in our experience. Because of unavoidable wheel slip, however, odometry drift over hundreds of meters of driving can cause large errors in vehicle motion estimation. An example is shown in Figure 4, where the blue curve represents the estimated path of the vehicle during the process of mapping data collection. In this example, the vehicle circulates a block twice after starting and stopping at the same place. One can see how odometry drift leads to each landmark being mapped to two very different locations. To correct for the effects of drift, we apply a technique known as loop closure, described later in this section.
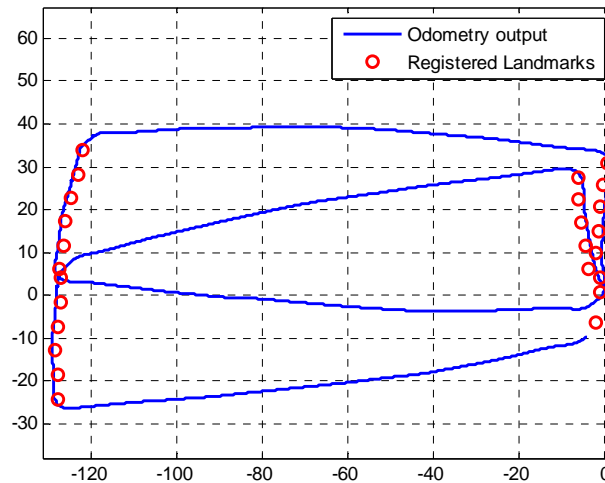

Figure 4. The first step in orchard mapping is to find the position of the vehicle via encoder odometry. In this example the vehicle circulates a block twice, starting and stopping at the same place. The resulting vehicle position (blue curve) and landmarks' positions (red circles) drifts as odometry errors accumulate.

Point Cloud Registration and Landmark Detection

Using the vehicle position obtained via encoder odometry, we register the lidar point cloud into a fixed frame with respect to the ground. For simplicity, the starting point of the vehicle is selected as the origin of the coordinate system. Figure 5 gives an example of the registered point cloud, color coded by elevation. The green points belong to the ground and the magenta points belong to the trees. The point cloud contains 3D structure information of the block and is used for both tree row and landmark detection.

Recall that the landmarks are made of retro-reflective tape; to locate them, we find the clusters with highest intensity within the point cloud. The yellow rectangle in Figure 5 shows an example of landmark detected in the lidar point cloud. Let d be the distance from a detected landmark to the vehicle, and let β be the orientation of the landmark with respect to the vehicle. The following equations are used to register the landmark with respect to the world coordinate frame:

$$x_l = x_v + d \cos(\theta + \beta)$$

$$y_l = y_v + d \sin(\theta + \beta)$$

where (x_l, y_l) are the coordinates of the landmark, (x_v, y_v) is the current position of the vehicle, and θ is the current vehicle heading.
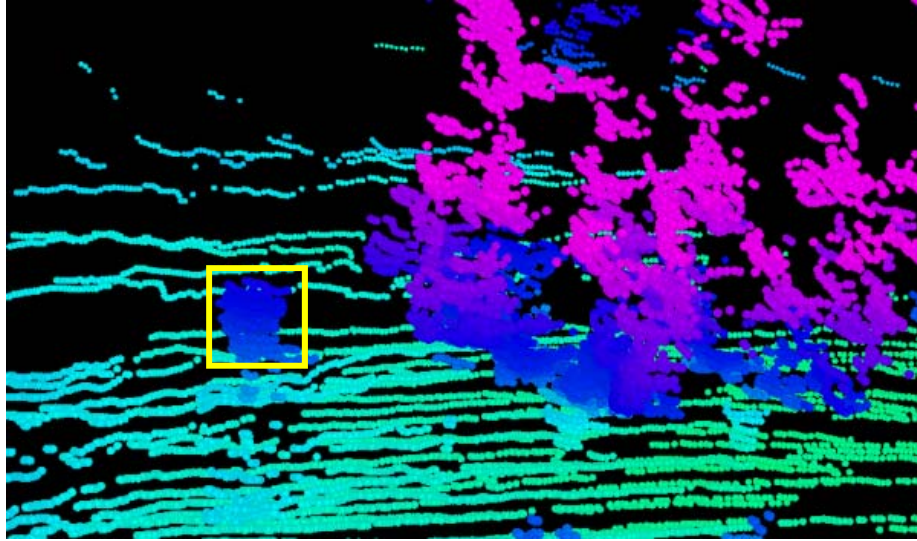
Figure 5. An example of lidar point cloud containing a landmark, labeled by the yellow rectangular. Points from landmarks contain high intensity and can be easily detected.

Loop Closure

This section discusses how to generate the map using the encoder odometry output and the detected landmarks. The landmarks are registered using encoder odometry in a fixed coordinate system with respect to the ground, shown as the red circles in Figure 4. To facilitate the mapping, we require the vehicle to start at a corner of the block, as indicated by the green arrow on the bottom right corner in Figure 6, and finish at the same corner, indicated by the blue arrow. This ensures the lidar sees the same landmark at the beginning and end of the data collection, a technique known as loop closure. Furthermore, if the vehicle circulates the block more than once, we also know that each landmark's appearance should be in the exact same location. Using this knowledge as constraints, we can post-optimize the encoder odometry by smoothing the vehicle's path such that the landmarks are aligned. Let γ_x and γ_y denote the gap between the first and the last landmark in the x- and y- directions, respectively, and γ_θ denote the angular gap, estimated by the encoder odometry. The following equations correct the odometry drift such that the first and the last landmark are aligned:

$$c_x = \gamma_x \frac{S'}{S}, \qquad c_y = \gamma_y \frac{S'}{S} \qquad c_\theta = \gamma_\theta \frac{S'}{S}$$

where c_x and c_y are the corrections applied to a landmark in the x- and y- directions, respectively, and c_θ is the angular correction. S is the overall driving distance around the block, and S' is the driving distance from the first landmark to the current landmark. Both S and S' are measured by the encoder odometry. As a result of loop closure, we can obtain a map containing coordinates of each landmark.
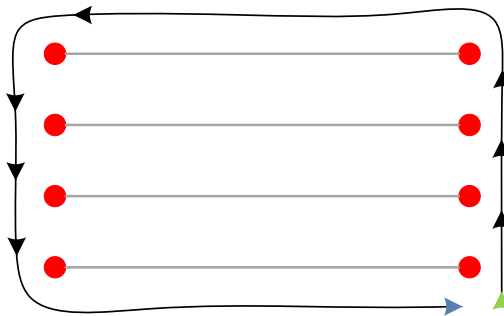

Figure 6. Illustration of vehicle trajectory used for orchard block mapping. The vehicle starts at the green arrow and stops at the blue arrow, both on the bottom right. This ensures that the lidar looks at the same landmark at the beginning and end of the data collection, and introduces a loop closure constraint to correct the odometry drift seen in Figure 4.

Dealing with Irregular Terrain

Encoder odometry and loop closure generate accurate orchard block maps most of the time. There are occasions, however, when excessive odometry drift—for example, when the terrain is overly irregular or steep, as Figure 7—cannot be corrected by loop closure alone. In these cases, we add landmarks to the four corners

of the block, such that the line segment formed by each additional landmark and its closest regular one is perpendicular to the tree rows (see Figure 7 for an illustrative example, where red circles represent the original landmarks and the yellow circles represent the extra ones). The extra landmarks provide four additional geometric constraints that are used to correct the odometry drift. More specifically, we map the landmarks on one side of the block at a time by forcing the line segments connecting the extra landmarks and their closest original one to be parallel. All other original landmarks are then placed between the extra ones using encoder odometry, and finally the landmarks on both sides of the block are placed on the map. We have used this method to successfully map blocks with very steep headlands that cause significant wheel slip, and that could not be mapped by encoder odometry and loop closure alone. An example of such block is shown in Figure 7.

Figure 7. Left: Example of steep headland that causes significant wheel slip and where the normal mapping method fails. In these situations we install four additional landmarks, represented by the yellow circles on the right, that introduce four new geometric constraints to the mapping process. The additional landmarks have extended the range of blocks that can be mapped and therefore the range of blocks in which the autonomous orchard vehicle can operate.

Experimental Results

The mapping method has been tested in two commercial orchards in Wexford and Bedford, PA. Figure shows two representative results. The left figure corresponds to the block in Figure 1, which is mostly flat and easy to map. The right figure corresponds to the block in Figure 7, in which we had to use four additional landmarks to deal with the slopes. In Figure 8, the map built by the method proposed here is represented by the red circles, with the gray lines representing the tree rows. The blue crosses represent the true positions of the landmarks, provided by a very high accuracy Applanix GPS-assisted inertial navigation system. For the block in Figure 1 (left side of Figure 8), the mean distance error between the mapped landmarks and the true ones is 19 cm. For the block in Figure 7 (right side of Figure 8), the mean distance error is 26 cm. We have found that these errors are small enough to warrant the maps can be used for autonomous navigation.
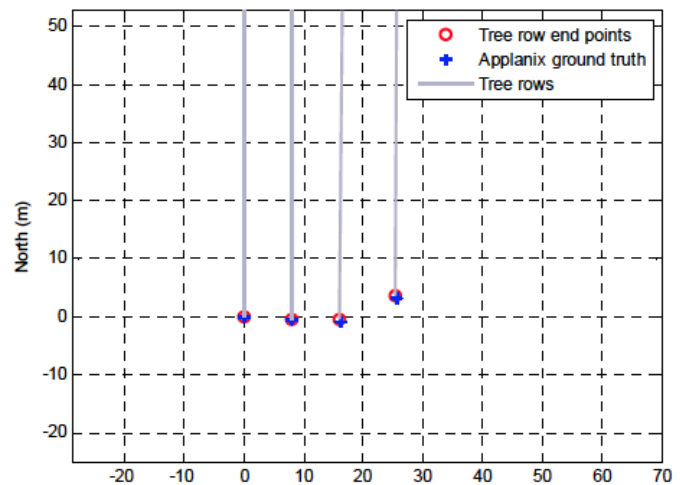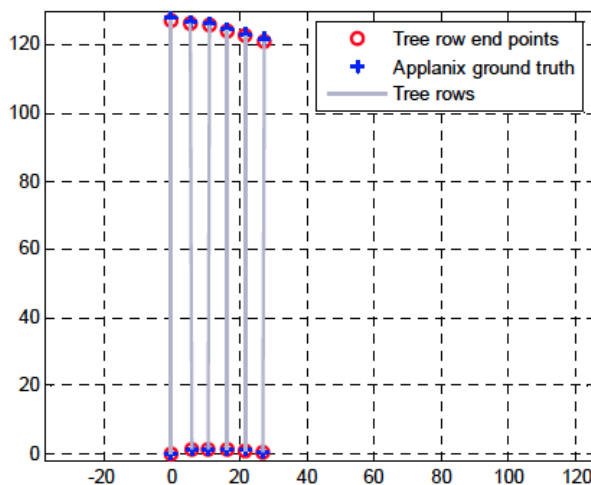

Figure 8. Orchard block maps obtained with the method proposed in this paper. On the left is a map of the block in Figure 1, and on the right is a map of the block in Figure 7. The red circles represent the output of our method; the blue crosses represent the true landmark positions provided by a highly accurate Applanix GPS-assisted inertial navigation system.

The maps shown above have been used in support of autonomous block coverage. Figure 9 shows an example for the block in Figure 1 (also Figure 8, left). In Figure 9, the red line represents the position of the vehicle calculated by the odometry-based localization system (Zhang et al., 2013), and the blue line represents the true position given by the Applanix system. One can see that the localization output matches very well the true trajectory, a result made possible in part because an accurate map is available. Figure 10 shows the histograms of the longitudinal and lateral position errors when the vehicle traverses five orchard rows. The maximum longitudinal error is 80 cm over 110-m rows, with the median error at 30 cm. The maximum lateral error is 50 cm, with the median error at 15 cm, in 5-m wide rows.
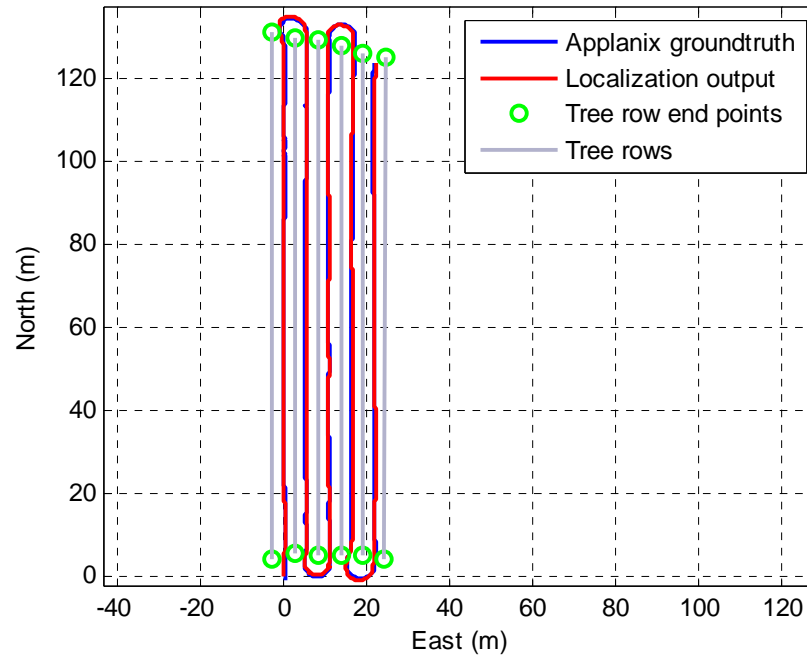

Figure 9. Autonomous block coverage using the map in Figure 7. The red line represents the position of the vehicle calculated by the odometry-based localization system and the blue line represents the true position given by the Applanix system. Over 90% of the time the vehicle is within 50 cm of the desired position.
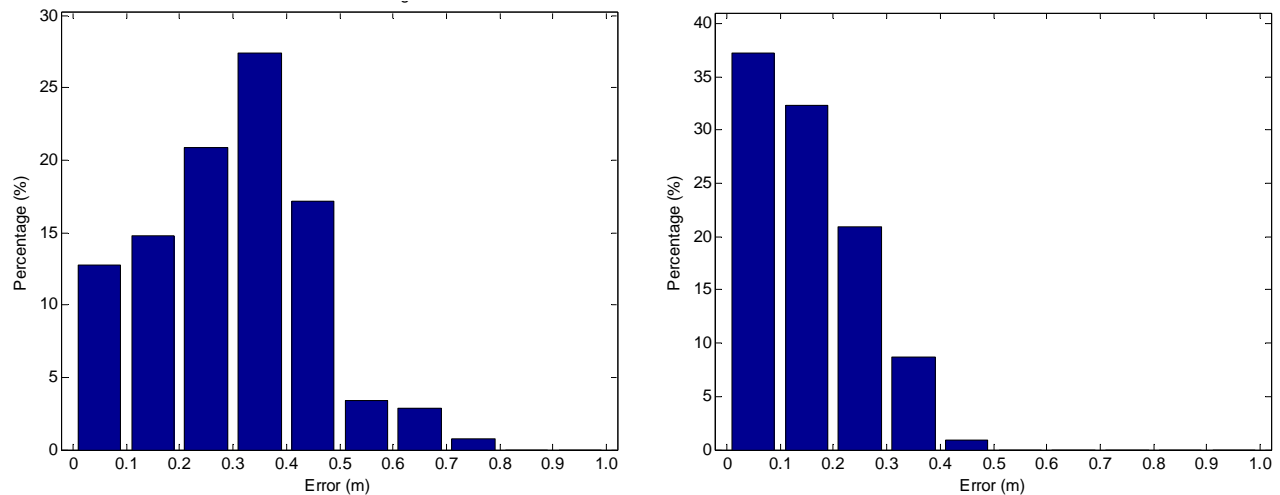

Figure 10. Histogram of longitudinal (left) and lateral errors (right) during the 5-row autonomous block coverage in Figure 9. The small overall position errors are due in part to the availability of an accurate orchard map.

Conclusion and Future Work

The orchard block mapping method proposed in this paper has been used to map several commercial apple orchards and therefore to enable many kilometers of autonomous block coverage. The method works in blocks with rows parallel to each other, which represent a large percentage of blocks in major apple producing states. With the addition of four extra landmarks, it also works well when the terrain induces significant wheel slip. For blocks with non-parallel rows, intermediate landmarks may be installed along the tree rows to provide additional information for mapping. Our future work will focus on augmenting the encoder odometry with laser-based odometry in an attempt to enable landmark-free mapping and navigation.

References

Astrand B. & Baerveldt, A. (2005). A vision-based row-following system for agricultural field machinery. Mechatronics, vol. 15, no. 2, pp. 251–269, 2005.

Bakker, T., Wouters, H., Asselt, K., & et al. (2008). A vision-based row detection system for sugar beet. Computers and Electronics in Agriculture, vol. 60, pp. 87–95, 2008.

Biber, P., Weiss, U., Dorna, M, & et al. (2012). 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, Oct. 2012.

Fischler, M. & Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

Hiremath, S., Evert, F., Heijden, G., & et al. (2012). Image-based particle filter for robot navigation in a maize field. Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal, Oct. 2012.

Libby, J., & Kantor, G. (2011). Deployment of a Point and Line Feature Localization System for an Outdoor Agriculture Vehicle. 2011 IEEE International Conference on Robotics & Automation (ICRA 2011), May, 2011.

Morrehead, S., Gilmore, B., Dima, C., & et al. (2012). Automating orchards: A system of autonomous tractors for orchard maintenance. Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal, Oct. 2012.

Roth, S. & Batavia, P. (2002). Evaluating path tracker performance for outdoor mobile robots. Automation Technology for Off-Road Equipment, July 2002.

Satow, T., Matsuda, K., Ming, S., & et al. (2004). Development of laser crop row sensor for automatic guidance system of implements. Conference on Automation Technology for Off-road Equipment, Kyoto, Japan, Oct. 2004.

Zhang, J., Chambers, A., Maeta, S., Bergerman, M., & Singh, S. (2013). 3D Perception for Accurate Row Following: Methodology and Results. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS). Tokyo, Japan, Nov 2013.