

A Practical Obstacle Detection System for Autonomous Orchard Vehicles

Gustavo Freitas, Bradley Hamner, Marcel Bergerman and Sanjiv Singh

Abstract—Safe robot navigation in tree fruit orchards requires that the vehicle be capable of robustly navigating between rows of trees and turning from one aisle to another; that the vehicle be dynamically stable, especially when carrying workers; and that the vehicle be able to detect obstacles on its way and adjust its speed accordingly. In this paper we address the latter, in particular the problem of detecting people and apple bins in the aisles between rows. One of our requirements is that the obstacle avoidance subsystem shouldn't add to the robot's hardware cost, so as to keep the acquisition cost to growers as low as possible. Therefore, we confine ourselves to solutions that use only the sensor suite already installed on the robot for navigation—in our case, a laser scanner, low-cost inertial measurement unit, and steering and wheel encoders. Our methodology is based on the classification and clustering of registered 3D points as obstacles. In the current implementation, obstacle avoidance takes in 3D point clouds collected in apple orchards and generates an off-line assessment of obstacle position. Tests conducted at our experimental orchard-like environment in Pittsburgh and an actual apple orchard in Washington state indicate that the method is able to detect people and bins located along the vehicle path. Stretch tests indicate that it is also capable of dealing with objects as small as 15 cm tall as long as they aren't covered by grass, and to detect people crossing the aisles at walking speed.

I. INTRODUCTION

Tree fruit production is a very labor-intensive business. In the US, for example, labor account for over 50% of the variable costs to produce apples. Additionally, the number of workers required varies significantly throughout the year—in the state of Washington, for example, it fluctuates between 5,000 workers in the winter time to 35,000 at the peak of harvest. Clearly, there is an opportunity to introduce automation solutions into tree fruit production to lower labor costs, smooth out labor requirements, and increase production efficiency. This opportunity is compounded by the introduction of high-density planting architectures in the past twenty years, where fruit grows along “walls” formed by the branches of trees just four to six feet apart. Autonomous vehicles driving down along these fruit walls can mow and spray, as well as carry workers pruning, thinning, performing tree maintenance, and harvesting.

For the past three years we have been developing a family of such vehicles, which we call Autonomous Prime Movers,

*This work is supported by the USDA National Institute of Food and Agriculture, Specialty Crop Research Initiative, under award no. 2008-51180-04876.

G. Freitas is with the Dept. of Electrical Eng., COPPE/Federal University of Rio de Janeiro, Brazil, funded by the Brazilian National Council for Research and Development (CNPq).

B. Hamner, M. Bergerman and S. Singh are with the Field Robotics Center, Robotics Institute, Carnegie Mellon University in Pittsburgh, PA, USA.

or APMs (Figure 1). The current APMs are capable of autonomously driving between a row of trees, turning at the end of the aisle and entering the next one. Row following is conducted at the center of the aisle (e.g., for sensing or mowing) or at a predefined distance from the trunk line (e.g., for pruning, thinning, or spraying). To be affordable to growers, the APMs do not carry a high-accuracy GPS-assisted inertial navigation system (INS), as is usual in agricultural automation. Rather, they navigate using only one laser rangefinder, a low-cost inertial measurement unit (IMU), and steering and wheel encoders.



Fig. 1. Two vehicles in the Autonomous Prime Mover family. (Left) Orchard workers thinning green fruit from onboard an autonomous platform. (Right) Workers using the autonomous vehicle as a “bin dog” to increase harvest efficiency.

Between 2009 and 2011 the five vehicles in the APM family drove a combined 330 km in research and commercial orchards in several US states. Preliminary results indicate that workers on an APM-mounted platform can conduct some tasks on the top of the trees in half the time taken by workers on ladders or on foot [7], [8].

While performance when driving between rows is satisfactory, the current system does not include the capability to detect obstacles in the aisle and adjust vehicle speed accordingly. Clearly, this is a safety requirement that must be addressed before APMs can become part of the tree fruit grower toolbox. Additionally, any obstacle detection system must not add to the hardware cost of the vehicle, lest cost issues increase the adoption barrier. Finally, the system must robustly detect the two major obstacles found in orchards: people and bins.

In this paper we address obstacle detection for autonomous orchard vehicles driving at working speeds of up to 1 m/s, using only the sensing suite already in place for row following—namely, a laser scanner, IMU, and steering and wheel encoders. Our methodology consists of four steps:

first, 3D range measurements are taken by the onboard laser scanner; second, the measurements are registered to the vehicle and inertial coordinate frames; third, the 3D points are classified according to their belonging (or not) to an obstacle; and fourth, the 3D points are clustered and the obstacle position is estimated. Obstacle avoidance is currently implemented as an off-line Matlab process. Tests conducted at our experimental orchard-like environment and an actual apple orchard in Washington state indicate that the method is able to detect people and bins placed on vegetation along the vehicle path. We also conducted “stretch” tests to verify how far beyond the original requirements the system can respond. The results indicate that it is capable of dealing with objects as small as 15 cm tall as long as they aren’t covered by grass, and to detect people crossing the aisles at walking speed.

The paper is organized as follows. In Section II we present a review of related work, with special focus on obstacle detection for ground vehicles operating in orchard-like agricultural environments. In Section III we present the autonomous orchard vehicle used as a platform in which the obstacle avoidance method was implemented and tested. In Section IV we present the method per se, and in Section V the results of extensive field experiments conducted in orchards. We conclude the paper in Section VI indicating avenues for future work.

II. LITERATURE REVIEW

Obstacle detection is a key capability for autonomous vehicles, and a specially important one for vehicles navigating in agricultural environments. In essence, obstacle avoidance consists of determining whether the space ahead of the vehicle is clear from obstructions for safe travel [15]. Its goal is to detect all obstacles along the path in time for the vehicle to react to them, while minimizing misclassifications.

Several obstacle detection approaches proposed and implemented in real systems, e.g., [3], [9], [14], showed good results for low speeds (under 3 m/s). Still, many difficulties are associated with obstacle detection in natural terrain, especially due to the presence of vegetation. On one hand, tall grass can be erroneously classified as an obstacle; on the other hand, an obstacle occluded by vegetation may not be detected. Additionally, we look here for technologies that are both practical (i.e., low-cost) and robust enough to augment the well-established navigation and driving systems.

Stereo vision [13], [19], omnidirectional vision [4] and color segmentation [2] have been used for obstacle detection and to determine the shape of the world around the vehicle. The main advantage of these approaches is their relatively low cost, as they rely essentially on commercial cameras. Passive vision, however, suffers from lighting, color constancy, and dynamic range effects that decrease the performance of the obstacle detection system and cause both false positive and false negative detections.

Many researchers have chosen to overcome these camera-based limitations by adopting active sensors such as sonars, radars [1], [10], and laser range finders. The laser is the

most recommended sensor for the application, due to critical advantages in accuracy, compared to radar, and processing speed, compared to vision systems [12].

Perhaps the most common solution for obstacle detection consists in installing the lasers on an active sweeping mechanism, e.g., as in [3], [6], [17]. By commanding the mechanism actuator, it is possible to change the orientation of the laser to measure the terrain from multiple perspectives, reducing effects of occlusion, mixed pixels and sunlight. This configuration, however, is impractical for our purposes, not only because it requires precise sweeping control—since sweeping orientation errors heavily influence the range measurements—but also because of its added cost, complexity, and failure mode.

We propose a push-broom configuration where the laser is installed in a fixed position on the vehicle, tilted to the horizontal, such that the measurement plane intersects the ground in a line at some distance ahead of the vehicle. As the vehicle moves forward, the laser measures different lines on the ground, allowing us to create point clouds representing the terrain surface while dispensing with sweeping mechanisms. As we show in the Results section, this simple and practical configuration is robust enough to meet the requirements of obstacle detection in orchard environments.

III. AUTONOMOUS ORCHARD VEHICLE

The base vehicle used in this work is the APM “Laurel” (Figure 2). It is based on the Toro MDE eWorkman electric utility vehicle, retrofitted to function either in manual or drive-by-wire mode. The base retrofitting process consisted on installing a steering motor, brake motor, motor drivers and steering and wheel encoders. Laurel is a research vehicle, where we implement and test orchard navigation technologies before they are ported to other vehicles in the APM family. It is important to note that, while Laurel is equipped with a high-accuracy Applanix POS 220 LV INS/GPS system, we do not use it for the obstacle detection described here—otherwise, it would be impossible to port the software to the other APMs, since they do not have such a system onboard. On the contrary, Applanix data is used for the sole purpose of calculating sensor measurement errors during the obstacle detection system development process. Likewise, while Laurel is equipped with a variety of laser scanners and cameras, here we only use one laser scanner. The relevant sensors for this work are: steering and wheel encoders with angular resolution of $0.38^\circ/\text{tick}$ and linear resolution of 2.33×10^{-5} m/tick; a CHR-6dm inertial unit from CH Robotics, one of the most inexpensive IMUs in the market; and a Sick LMS 291 laser scanner. The scanner has a 180° field-of-view with a 1° resolution, and a maximum scanning range of 80 m. The range resolution for measuring distances between 1-20 m is 10 mm, with 35 mm accuracy and 10 mm standard deviation (1σ).

The main onboard processor is a rugged, waterproof, industrial computer with an Intel Core 2 Duo 1.6 GHz CPU with 4GB DDR2 DRAM from Small PC. The navigation



Fig. 2. Laurel Autonomous Prime Mover. In this work we use only the steering and wheel encoders, low-cost IMU (blue box on the bottom right) and one laser scanner (green box on top right) for obstacle detection.

software runs on Ubuntu Linux, with the message passing provided by Willow Garage’s Robot Operating System (ROS).

The laser is installed in a push-broom configuration, with the inclination with respect to the horizontal defined according to a safety buffer that we wish to maintain ahead of the vehicle. Laurel travels at a maximum speed of 2 m/s, and has a stopping time of one to two seconds. Therefore we choose to detect obstacles at least 4 m away to avoid abrupt maneuvers. Since the maximum possible mounting height is 1.45 m above the ground, we installed the laser scanner at an angle of -20° with respect to the horizontal.

To register the 3D measurements from the laser scanner we define three coordinate systems as shown in Figure 3. In the following, a superscript letter represents the frame with respect to which a quantity is described. The laser frame O^L corresponds to the sensor measurement plane. The vehicle frame O^V is located at the vertical projection on the ground of the center of the rear axle. The inertial frame O^I coincides with the vehicle’s initial pose.

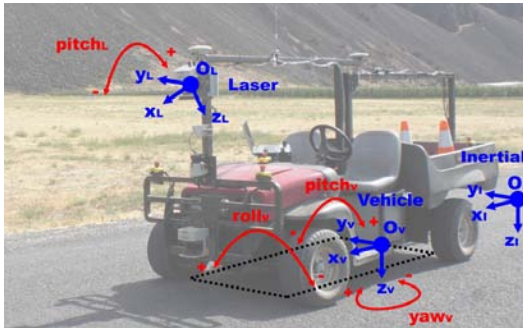


Fig. 3. Laser scanner, vehicle, and inertial coordinate frames.

To find the coordinate transformation between O^L and O^V , we follow the technique proposed in [18]. The idea consists on calculating the translational and rotational offsets between the laser and the vehicle by taking measurements of an environment of known geometry, in our case a single vertical pole on flat ground covered in reflective tape (Figure 7). The result obtained is a translation $t_L^V = [2.61, -0.04, -1.45]^T$ m and an orientation expressed by roll, pitch and yaw angles $\phi_L^V = [0.05^\circ, -21.25^\circ, -0.56^\circ]^T$.

IV. OBSTACLE DETECTION METHODOLOGY

We propose a practical obstacle detection system for autonomous orchard vehicles, i.e., one that does not add hardware cost to the vehicle and does not depend on expensive, high-accuracy GPS-based localization. The system takes as inputs the 3D laser scanner, IMU, and encoders measurements and outputs a binary assessment of obstacles on the vehicle’s path. Obstacle detection is performed in four steps: sensing and sensor data filtering, registering the 3D laser scanner measurements with the vehicle and inertial coordinate frames, classifying the registered 3D points as candidate obstacles, and clustering candidate points as obstacles. In its present form the system is implemented as an off-line Matlab process that runs on actual field data collected in apple orchards. In its final form it will run on-line to adjust the vehicle speed in reaction to detected obstacles, and alert the operator as to their presence.

A. Sensing and Sensor Data Filtering

The first step in obstacle detection consists of taking measurements from the laser scanner, IMU, and encoders, and appropriately filtering them to account for various sources of errors.

For the laser scanner, the main error source is the so-called “salt and pepper” noise [12], measurements that do not belong to the local neighborhood and do not obey the local surface geometry [16]. Sources of “salt and pepper” noise include boundaries of occlusions, surface reflectance, and multi-path reflection. We adopt here an outlier rejection filtering technique similar to the one employed in [18]. Each laser scanner array l_i containing the range r_{ij} of 181 points is modeled as a continuous function with mean μ and standard deviation σ . A point is identified as an outlier if it lies beyond two standard deviations from the mean:

$$|r_{ij} - \mu| > 2\sigma \Rightarrow r_{ij} \text{ is an outlier} \quad (1)$$

Outliers are then replaced by the arithmetic mean of its neighboring points:

$$r_{ij} = \frac{r_{ij-1} + r_{ij+1}}{2} \quad (2)$$

This algorithm, of course, does not eliminate “swell-behaved” outliers that “hide” within 2σ from the mean. It is just employed to remove points that are clearly out of the measured environment scope.

For the encoders, filtering is used to compute vehicle speed and acceleration. We use a low-pass filter currently implemented by the Matlab function:

```
filter(ones(1,windowSize)/windowSize, 1, inputData);
```

with a window size of 20 measurements. A typical result is shown in Figure 4.

Finally, for the low-cost CHR-6dm IMU, the main source of error is the assumption made by the IMU internal extended Kalman filter (EKF) that the accelerometers only measure the gravity vector. Because of that, the errors associated

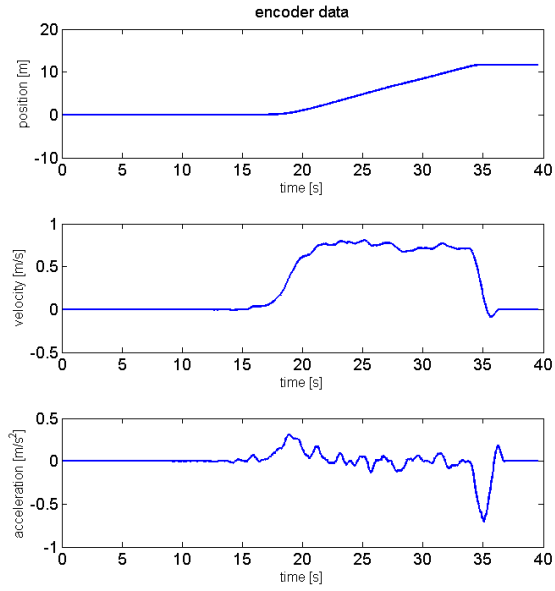


Fig. 4. Vehicle position, velocity and acceleration obtained with encoder odometry and low-pass filtering.

with pitch angle measurement increase whenever the vehicle accelerates. The maximum absolute error measured is about 5° (Figure 5). Noting that the pitch error curve in Figure 5 is similar to the vehicle acceleration obtained via encoder odometry, it is possible to cancel the effect of vehicle acceleration a in the measured pitch angle ϑ_{IMU} via:

$$\vartheta = \vartheta_{IMU} - 5.02a \quad (3)$$

where the gain factor 5.02 was found via a linear fit of the errors between the Applanix and CHR-6dm IMU obtained during sixteen field tests. After the correction, the maximum absolute value of the corrected pitch angle error ϑ is about 0.5° .

B. Registering 3D Laser Scanner Measurements

The laser scans the terrain in front of the vehicle, providing range to points in the sensor measurement plane. These range measurements are registered with the vehicle's and the inertial coordinate frames for the purposes of informing the vehicle of obstacles in its way. The registration process consists of two coordinate transformations, from O^L to O^V and then to O^I .

The first transformation represents the measured points with respect to the vehicle frame p_L^V . For that, we use the laser pose (ϕ_L^V, t_L^V) obtained with the calibration procedure. The second transformation describes the vehicle coordinate with respect to the inertial frame, located at the vehicle's initial position. The vehicle orientation is obtained with respect to vertical and horizontal planes, and the yaw angle (ψ_V^I) is discarded. Roll and pitch angles $(\varphi_V^I, \vartheta_V^I)$ are measured by the CHR-6dm IMU, and the vehicle translation t_V^I is obtained using encoder odometry. The 3D laser points with respect to the inertial frame p_L^I are given by:

$$p_L^I = A_V^I A_L^V p^L = A_L^I p^L \quad (4)$$

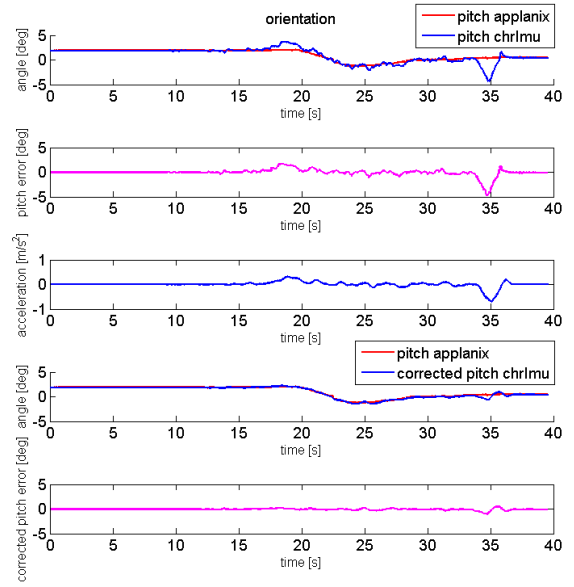


Fig. 5. From top to bottom: vehicle pitch angle obtained with the CHR-6dm in blue and with the high-accuracy Applanix in red; pitch measurement error; acceleration obtained via encoder odometry; pitch angle from the CHR-6dm corrected using the measured acceleration; corrected pitch error.

where A_j^i is the homogeneous transformation from frame O^j to frame O^i . Figure 6 shows a typical example of 3D laser data when the vehicle approaches a wooden box placed along its path.

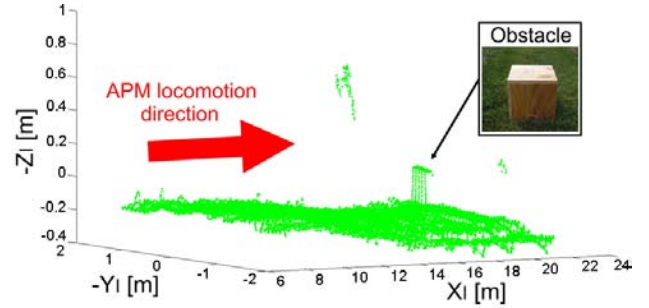


Fig. 6. Three-dimensional laser data in the inertial frame when the vehicle approaches a wooden box.

At this point of the work, we are not considering uncertainties related to laser measurements. It has not been a problem so far, considering the target obstacles dimension and the sensor accuracy. Another simplification consists on disregarding wheel encoders drifting errors. When integrated to the vehicle's driving software, the obstacle detector will employ the pose (t_V^I, ψ_V^I) provided by the APM localization system described in [11].

C. Point Clouds Classification

The next step is to analyze the registered 3D data and classify them as belonging to terrain or obstacles. We employ a gradient-based classification methodology similar to the ones in [3], [5].

The method essentially consists of searching for irregularities between adjacent points measured by the laser scanner. Two neighboring points $p_1 = [x_1, y_1, z_1]^T$ and $p_2 = [x_2, y_2, z_2]^T$ are classified as candidate obstacles if:

$$\frac{|z_2 - z_1|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} > \tan \alpha \quad (5)$$

or

$$|z_2 - z_1| > h \quad (6)$$

where α is a reference angle and h is a reference height employed for discontinuity detection. Figure 7 illustrates how the height and angle are calculated.

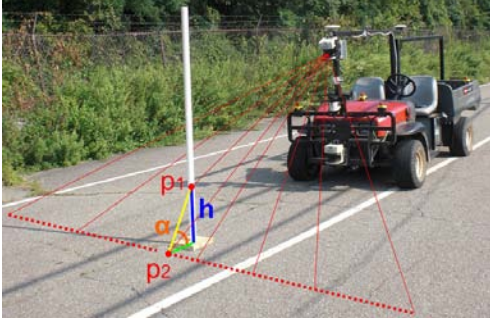


Fig. 7. Gradient method for classifying irregularities. Points are deemed candidate obstacles if the height or angle between them and adjacent points are greater than given thresholds h and α , respectively.

The classification is executed in two directions: first we look for lateral discontinuities along the laser measurement plane; then for longitudinal discontinuities along the vehicle's direction of motion. The former depends only on the laser's current measurement vector; the latter depends on the analysis of multiple vectors along a period of time, stored in short-term memory.

The classification reference variables α, h indicate the local geometry irregularity expected between the environment and obstacle. Considering the target obstacles geometry and dimensions, the values used for gradient classification are $\alpha_{lat} = 18^\circ$ and $h_{lat} = 0.2$ m in the lateral direction and $\alpha_{long} = 35^\circ$ and $h_{long} = 0.15$ m in the longitudinal direction. Figure 7 shows an example of lateral and longitudinal discontinuities between ground and a wooden box of side 30 cm.

Once lateral and longitudinal discontinuities are identified, 3D points that belong to both categories are marked as obstacle edges. This is useful because many obstacle avoidance approaches operate based on finding edges of objects in the vehicle path [15].

D. Clustering Obstacle Candidates

The last step in obstacle detection is to cluster the candidate obstacle points to eliminate false positives caused by terrain irregularities and vegetation. The idea here is to declare as an obstacle only those clusters consisting of a minimum number of points.

The clustering procedure is divided in three stages. First we cluster the obstacle edges. Second, we cluster the other

candidates points that correspond to the obstacle body. Finally, we combine the clusters in order to estimate the obstacle position.

Consider as an example the wooden box in Figure 8. The edges are composed by two point sets forming red vertical lines. We begin the clustering process combining the points forming these vertical lines. All the points (pe) identified as edges are clustered together if the distance between them is smaller than a threshold d_e :

$$|pe_i - pe_{i+1}| < d_e \Rightarrow pe_i, pe_{i+1} \in cluster_i \quad (7)$$

Here we use $d_e = 0.1$ m. The clusters with more elements than a given number $c_e \in \mathbb{N}$ represent the obstacles edges:

$$num(cluster_i) > c_e \Rightarrow cluster_i = obstacleEdge_i \quad (8)$$

The specification of the value c_e depends on the characteristics of the environment. For terrains with low grass and sparse vegetation, we found that $c_e = 4$ provides good results. If the terrain is covered by high grass and dense vegetation, this value can reach up to $c_e = 20$.

A similar procedure is executed for the other, non-edge points classified as candidate obstacles. These discontinuities correspond to obstacle body points (pb), that are clustered together if the distance between them is smaller than a threshold d_b :

$$|pb_j - pb_{j+1}| < d_b \Rightarrow pb_j, pb_{j+1} \in cluster_j \quad (9)$$

Here we use $d_b = 0.6$ m. Additionally, a cluster represents the obstacle body if the number of elements in it is larger than a given number $c_b \in \mathbb{N}$:

$$num(cluster_j) > c_b \Rightarrow cluster_j = obstacleBody_j \quad (10)$$

where c_b value ranges from 50 to 200.

The obstacle position is found by combining the clusters representing the obstacle's edges and body. The process is executed by a sweeping algorithm using the obstacle body clusters. The algorithm evaluates the distance from body clusters to all edge clusters. If the distance is smaller than a threshold d_o , it is assumed that the body and edge clusters represent the same obstacle:

$$|obstacleBody_j - obstacleEdge_i| < d_o \Rightarrow obstacleBody_j, obstacleEdge_i \in obstacle_j \quad (11)$$

The threshold employed is $d_o = 0.6$ m.

Finally, the obstacle's planar position corresponds to the mean value of the associated edges:

$$\begin{bmatrix} obstacle_{j_x} \\ obstacle_{j_y} \end{bmatrix} = \begin{bmatrix} \mu(obstacleEdge_{i_x}) \\ \mu(obstacleEdge_{i_y}) \end{bmatrix}, \quad (12) \\ \forall obstacleEdge_i \in obstacle_j$$

Note that the obstacle’s body clusters are used solely to identify the obstacle, reducing the number of false positives, and do not influence the estimation of its position. The procedure is illustrated in Figure 8, with the detected obstacle marked as a black star.

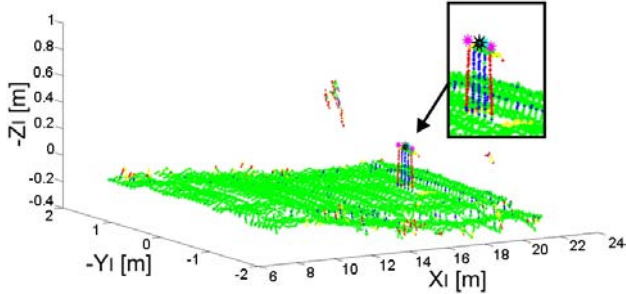


Fig. 8. Three dimensional points representing the terrain profile and obstacle. The black star marks the detected obstacle, found from a combination of edges (magenta starts) and body clusters (cyan starts). The lateral irregularities are plotted in yellow and the longitudinal irregularities are plotted in blue. The obstacle edges are plotted in red.

V. EXPERIMENTAL VALIDATION

To assess the feasibility of the obstacle detection methodology presented in Section IV, we collected field data from an experimental orchard-like nursery and an actual apple orchard. The former is a half-acre ornamental tree nursery planting at Robot City, a robotics experimentation site in Pittsburgh, PA. The latter is Washington State University’s Sunrise Orchard in Rock Island, WA. At both locations we manually drove the vehicle as it approached a person and a commercial apple bin, and recorded the registered 3D laser scanner, encoder, and IMU data. In Pittsburgh we collected additional data while a person walked in front of the vehicle; and at Sunrise we collected 3D data as the vehicle drove toward a set of wooden boxes of sizes 30 cm, 25 cm, 20 cm, and 15 cm, both in short and tall grass. The objective with these additional datasets was to stress-test the system and assess how well it can perform beyond the stated requirements of detecting stationary people and bins.

The data collected was processed by the obstacle detection methodology in Matlab. It takes as input the log files and outputs a three dimensional representation of the environment, a binary assessment of the presence of obstacles, and the obstacle’s location with respect to the inertial frame O^I .

A. Detecting People and Apple Bins

To conduct this experiment we used a commercial apple bin with dimensions $124 \times 116 \times 77$ cm and a person 1.72 m tall weighing 63 kg. The cluster parameters employed to identify the bin and the person edges and body are $c_e = 20$ and $c_b = 200$, respectively. In the results figures that follow, front view represents the actual image seen by the laser scanner; top view represents the $x - y$ projection of the 3D laser point cloud; and isometric view represents the 3D

representation of the point cloud with elevation = 35° and azimuth = -50° .

Figure 9 presents an example of bin detection. Due to its large dimensions, the bin is recognized as several “obstacles”. This is irrelevant in our operation because the vehicle will react to the closest obstacle presented to it by the detection system. The system correctly detected the bin in all of six experiments conducted.

Figure 10 presents an example of the system detecting a person standing in between the rows of trees. Here again the person was correctly detected in all of ten experiments conducted.

B. Pushing the Performance

Once the requirements to detect people and bins in orchards were met, we turned our attention to studying the obstacle detection system’s operational limits. In particular, we were interested in finding out what’s the smallest identifiable obstacle, and what’s the system’s response to moving obstacles.

1) Dealing with Small Obstacles:

To test the system’s response to obstacles smaller than an apple bin, we built wooden boxes with sides 30, 25, 20, and 15 cm. They were placed in two blocks along the aisles of the Sunrise Orchard. In the first block the grass was 5 cm tall and, in the second, 17.5 cm tall. In the first block the clustering parameters used are $c_e = 4$ and $c_b = 50$. Figure 11 shows a typical result, with the system capable of correctly separating the five obstacles.

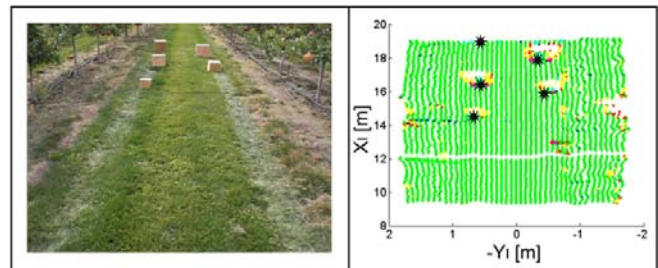


Fig. 11. To push the performance limits of the obstacle detection system we ran it on 3D data collected with five wooden boxes in between the rows of trees at Sunrise Orchard. The boxes measure 30, 25, 20, and 15 cm on the side. The obstacles are correctly detected, as shown by the black stars on the top view graph on the right.

In the tall grass block, as expected, the results are not as accurate. In this case the standard cluster parameters are $c_e = 10$ and $c_b = 150$. The obstacle detection system is able to correctly identify the 30 and 25 cm cubes (Figure 12).

The smaller cubes with 20 and 15 cm sides are partially occluded by the tall grass, and the detection system is not able to distinguish them from the vegetation. If we decrease the cluster parameters c_e and c_b to increase the system’s sensitivity, the algorithm starts presenting false positives before detecting the boxes. Figure 13 presents a typical result obtained with the 20 cm box, where one can see the three false obstacles marked by black stars.

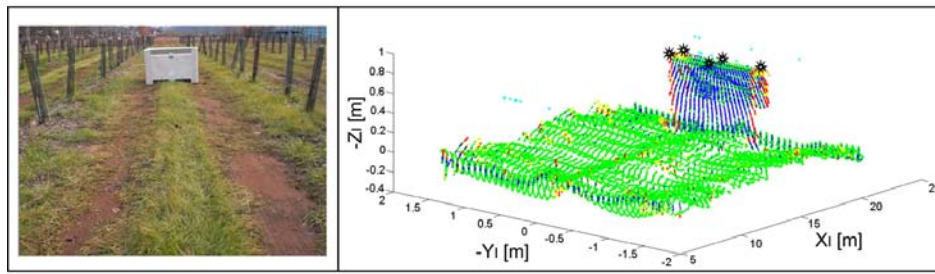


Fig. 9. Apple bin detection at Robot City. The detection system finds five “obstacles”, marked by black stars on the isometric view graph on the right.

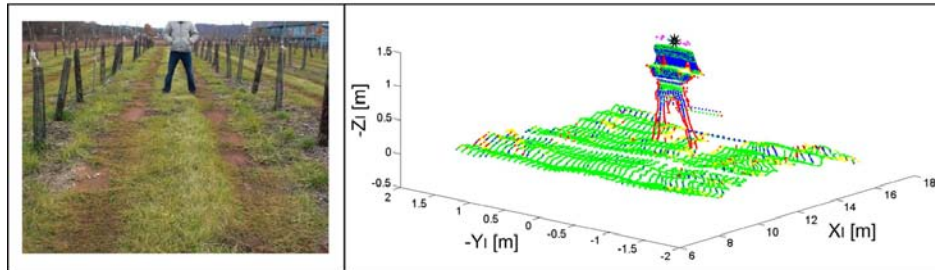


Fig. 10. Person detection at Robot City. The detection system marks the perceived obstacle by a black star on the isometric view graph on the right.

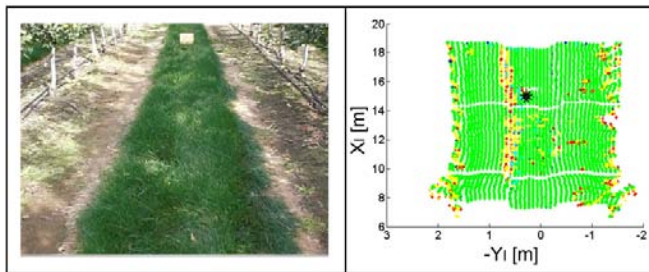


Fig. 12. Detecting a 25-cm wooden box in tall grass. This experiment is at the limit of the obstacle detection system’s capability to find the obstacle.

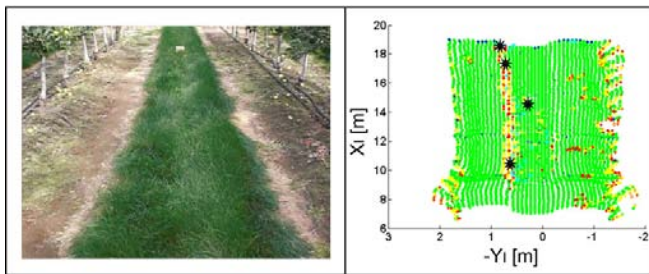


Fig. 13. The obstacle detection is not capable of uniquely identifying a 20 cm box in tall (17.5 cm) grass. In this example, the system found the box and three other false positives.

2) Dealing with Moving Obstacles:

The obstacle detection system, with its push-broom configuration, was designed for low-cost, stationary obstacle detection. In practice, however, we can always expect people moving in front of the vehicle. Therefore, it is important to check the system’s capacity to detect dynamic obstacles. To

that end we had a person perpendicularly crossing the path of the vehicle at walking speed. The result is presented in Figure 14. The detection system identifies several obstacle edges, according to the person’s motion. Due to the cluster distance d_o , the edges are grouped in one obstacle, marked by a black star in the graph.

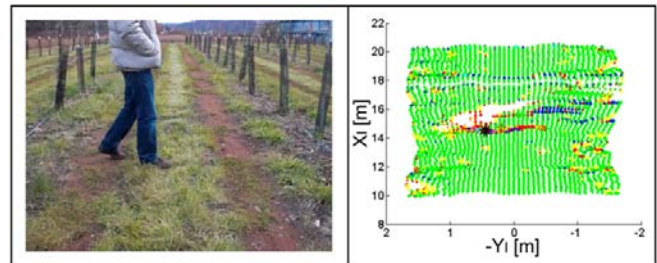


Fig. 14. A person crossing in front of the vehicle creates a dynamic point cloud from which the system extracts obstacle detection information. Here the system detects several object edges according to the worker’s motion; the edges are clustered together as one obstacle marked by a black star on the top view graph on the right.

VI. CONCLUSION AND FUTURE WORK

The obstacle detection system performed satisfactorily when using actual field data. At Robot City in Pittsburgh, both the bin and the person were scanned in 16 experiments from different orientations with no false positives. At Sunrise Orchard the system, designed to detect large bins, was able to detect boxes as small 15 cm tall on 5 cm grass, and 25 cm tall on 17.5 cm grass. Finally, back at Robot City the system was able to detect a person moving perpendicular to the vehicle’s path at walking speed. Of course, we do not

claim that this means the system can deal with any type of dynamic obstacle, but the result obtained is certainly a positive one.

By varying the cluster parameters c_e and c_b , it is possible to adjust the system's response. Decreasing the parameters will increase the system sensibility, allowing it to detect smaller obstacles, but also introducing a larger number of false positives. This undesirable effect is aggravated when dealing with irregular vegetation.

The obstacle detection system described in this paper is part of the larger goal of demonstrating an APM's feasibility to operate year-round in a commercial production environment. The methodology was tested and validated using real-world data from our experimental orchard-like nursery and a commercial orchard. Currently, the data processing is executed off-line in Matlab. There are several steps we need to follow before the system can be integrated into the APM's navigation system.

First and foremost, the obstacle detection method was developed and tested with data from a dedicated laser scanner mounted to meet the method's requirements. In an actual field deployment, that one laser scanner would have to provide data to both the navigation and the obstacle detection modules. The navigation module, however, currently uses a horizontally-mounted scanner about a meter off the ground. Therefore, we need to refactor it so it can operate with the laser mounted at a higher location in the push-broom configuration required for obstacle detection.

The second step is to integrate the obstacle detection system with the vehicle's driving system. The detection system will be implemented within the Robotics Operating System framework, to enable online obstacle detection and collision avoidance. Whenever an obstacle is detected in front of the vehicle, a message indicating the position of the object with respect to the vehicle must be generated. The autonomous driving system then uses this information to take appropriate action, including if necessary stopping the vehicle and alerting the operator. In the case of multiple obstacles, the detection system informs the position of the closest one to the vehicle. Once the operator removes the obstacle, the system resumes driving.

The integrated obstacle detector will present different characteristics from the ones presented here, caused by the new laser configuration and vehicle's pose provided by the APM localization system, among others. One important future requirement consists on obtaining the final sensor setup error covariance and consider uncertainty during 3D laser scanner measurements registration.

For the types of operations necessary in tree fruit production, the vehicle is usually navigating in between rows at low speeds (< 2 m/s). Because safety is a top priority in agricultural robotics, the obstacle detection system implementation must run sufficiently fast to process all data, identify the obstacles and stop the vehicle before a collision occurs. New field tests should be accomplished to validate that, emulating challenging conditions such as dealing with several obstacles moving in different directions.

REFERENCES

- [1] B. Barshan and R. Kuc, "A bat-like sonar system for obstacle localization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 4, pp. 636–646, 1992.
- [2] P. Batavia and S. Singh, "Obstacle detection using adaptive color segmentation and color stereo homography," in *IEEE International Conference on Robotics and Automation*, Seoul, 2001, pp. 705–710.
- [3] —, "Obstacle detection in smooth high curvature terrain," in *IEEE International Conference on Robotics and Automation*, Washington, 2002, pp. 3062–3067.
- [4] C. Cauchois, E. Brassart, L. Delahoche, and C. Drocourt, "Spatial localization method with omnidirectional vision," in *IEEE International Conference on Advanced Robotics*, Coimbra, 2003, pp. 287–292.
- [5] T. Chang, T. Hong, S. Legowik, and M. Abrams, "Concealment and obstacle detection for autonomous driving," in *Proc. of the Intl. Association of Science and Technology for Development-Robotics and Application*, 1999.
- [6] B. Hamner, S. Singh, S. Roth, and T. Takahashi, "An efficient system for combined route traversal and collision avoidance," *Autonomous Robots*, vol. 24, no. 4, 2008.
- [7] B. Hamner, M. Bergerman, and S. Singh, "Autonomous orchard vehicles for specialty crops production," in *ASABE Annual International Meeting*, Louisville, Kentucky, 2011.
- [8] —, "Specialty crop automation with autonomous vehicles," in *IEEE International Conference on Robotics and Automation*, St. Paul, Minnesota, 2012, pp. 1829–1835.
- [9] L. Henriksen and E. Krotkov, "Natural terrain hazard detection with a laser rangefinder," in *IEEE International Conference on Robotics and Automation*, Albuquerque, 1997, pp. 968–973.
- [10] R. Kuc, "A spatial sampling criterion for sonar obstacle detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 686–690, 1990.
- [11] J. Libby and G. Kantor, "Deployment of a point and line feature localization system for an outdoor agriculture vehicle," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 1565–1570.
- [12] K. Lingemann, A. Nchter, J. Hertzberg, and H. Surmann, "High-speed laser localization for mobile robots," *Robotics and Autonomous Systems*, vol. 51, no. 4, pp. 275–296, June 2005.
- [13] R. Mandelbaum, L. McDowell, L. Bogoni, B. Reich, and M. Hansen, "Real-time stereo processing, obstacle detection, and terrain estimation from vehicle-mounted stereo cameras," in *IEEE Workshop on Applications of Computer Vision*, 1998, pp. 288–289.
- [14] H. Schafer, A. Hach, M. Proetzsch, and K. Berns, "3D obstacle detection and avoidance in vegetated off-road terrain," in *IEEE International Conference on Robotics and Automation*, Pasadena, 2008, pp. 923–928.
- [15] S. Singh and P. Keller, "Obstacle detection for high speed autonomous navigation," in *IEEE International Conference on Robotics and Automation*, Sacramento, 1991, pp. 2798–2805.
- [16] S. Sotoodeh, "Outlier detection in laser scanner point clouds," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI, no. 5, pp. 297–302, 2006.
- [17] A. Stentz, C. Dima, C. Wellington, H. Herman, and D. Stager, "A system for semi-autonomous tractor operations," *Autonomous Robots*, vol. 13, no. 1, pp. 87–104, 2002.
- [18] J. Underwood, A. Hill, and S. Scheduling, "Calibration of range sensor pose on mobile platforms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, 2007, pp. 3866–3871.
- [19] C. Urmson, M. Dias, and R. Simmons, "Stereo vision based navigation for sun-synchronous exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Switzerland, 2002, pp. 805–810.