

A Pipeline for Trunk Detection in Trellis Structured Apple Orchards

Suchet Bargoti, James P. Underwood, Juan I. Nieto, and Salah Sukkarieh

Australian Centre for Field Robotics, The University of Sydney, NSW 2006, Australia

e-mail: j.underwood@acfr.usyd.edu.au, j.nieto@acfr.usyd.edu.au, s.sukkarieh@acfr.usyd.edu.au

Received 20 June 2014; accepted 4 February 2015

The ability of robots to meticulously cover large areas while gathering sensor data has widespread applications in precision agriculture. For autonomous operations in orchards, a suitable information management system is required, within which we can gather and process data relating to the state and performance of the crop over time, such as distinct yield count, canopy volume, and crop health. An efficient way to structure an information system is to discretize it to the individual tree, for which tree segmentation/detection is a key component. This paper presents a tree trunk detection pipeline for identifying individual trees in a trellis structured apple orchard, using ground-based lidar and image data. A coarse observation of trunk candidates is initially made using a Hough transformation on point cloud lidar data. These candidates are projected into the camera images, where pixelwise classification is used to update their likelihood of being a tree trunk. Detection is achieved by using a hidden semi-Markov model to leverage from contextual information provided by the repetitive structure of an orchard. By repeating this over individual orchard rows, we are able to build a tree map over the farm, which can be either GPS localized or represented topologically by the row and tree number. The pipeline was evaluated at a commercial apple orchard near Melbourne, Australia. Data were collected at different times of year, covering an area of 1.6 ha containing different apple varieties planted on two types of trellis systems: a vertical I-trellis structure and a Güttingen V-trellis structure. The results show good trunk detection performance for both apple varieties and trellis structures during the preharvest season (87–96 % accuracy) and near perfect trunk detection performance (99% accuracy) during the flowering season. © 2015 Wiley Periodicals, Inc.

1. INTRODUCTION

Information gathering and processing are becoming increasingly important in horticulture as farmers aim to optimize control processes and implement better farm management techniques. Accurate information about the crop, such as crop health, yield estimates, and tree counts, can help to efficiently target chemigation, fertigation, and fruit thinning processes, which ultimately leads to maximizing yield.

Farmers already gather information for precision farming, however it is a time-consuming and labor-intensive task. Additionally, the data are heavily subsampled and extrapolated, such that farmers use their judgment to pick a few trees that best represent the average over the entire farm. Recent advances in automation for field robotics have enabled us to start making key measurements for *all* trees in a timely and accurate manner. An appropriate information management system is therefore required, which can support high-resolution data on a farmwide scale. The data

could range from raw sensor output to higher-level analyses that evaluate flower and fruit counts. Appropriate storage and access to the information are beneficial for agronomists, farmers, and scientists for managing the orchard optimally.

A logical way to structure and manage an orchard information system is to quantize and associate data with the standard orchard unit, which is the individual tree. It enables all aspects of information processing to be done in a topological tree-by-tree fashion rather than relying on three-dimensional Cartesian mapping. For example, consider change detection for growth rates. We could attempt to visualize this in three dimensions over the entire farm, requiring accurate three-dimensional (3D) sampling, vehicle localization, and scan registration and alignment. Instead, a simpler approach is to estimate local properties in the data such as tree height, canopy volume, and yield count, and to link them to individual trees in the farm topology (e.g., row 21, tree 38 has 128 apples). Furthermore, operational tasks such as harvesting, thinning, and precision spraying can be customized to the individual trees based on this information.

Detecting individual trees is a key component of such an information system. This paper presents an autonomous

Direct correspondence to Suchet Bargoti: s.bargoti@acfr.usyd.edu.au



(a) The research ground vehicle *Shrimp* traversing between two V-trellis rows, scanning one side using a SICK LMS-291 2D lidar. Image taken during pre-harvest season.



(b) Apple trees planted in an I-trellis formation. Image taken during flowering season

Figure 1. Apple orchard at Warburton 3799, Victoria hosting a variety of apple types on two different trellis structures.

pipeline for the detection of apple trees in trellis structured orchards, using multimodal sensing. A lidar and a monocular RGB camera are mounted on an unmanned ground vehicle, where laser data are used to capture an initial estimate of trunk location and the model is updated through visual analysis of the images. The trunks, being primary components of the trees, can therefore be used to map tree locations. The pipeline is designed to be operated in a real orchard, and it was tested on an apple orchard located near Melbourne, Australia (Figure 1). The farm hosts two different variants of trellis structure, namely the standard vertical I-trellis formation and a newer Güttingen V-trellis plantation. Built with support poles and wires along the rows, these structures provide for better weight support of the tree limbs. They also allow for more sunlight for the fruits and easier harvesting (Christensen, 2012). The orchard also has a top cover netting to prevent damage to trees caused by hail.

The contribution of this paper is an end-to-end pipeline for detection of individual trees in a trellis structured apple orchard using lidar and image data. This includes adaptations from the segmentation framework proposed by Wellington and Campoy (2012) to enable operation in orchards with complex trellis structures and an extension of the work previously presented in Bargoti, Underwood, Nieto, and Sukkarieh (2014) to include multimodal sensing. Furthermore, the diversity of the pipeline is demonstrated through application over multiple trellis formations, apple varieties, and at different times of the year (crop maturity). Finally, this paper presents an analysis of the validity of the

observation models that are used to relate the multimodal observations to trunk detection.

The remainder of the paper is organized as follows. Section 2 presents related work on tree/trunk detection. Section 3 gives an in-depth description of the pipeline built for apple trunk detection. In Section 4 we evaluate the success of the pipeline on the orchard, discussing its capabilities and shortcomings. In Section 5 we discuss the significance of the results compared to the requirements of a complete orchard inventory system. We conclude in Section 6, discussing the future directions of this work.

2. RELATED WORK

Automation and advanced sensing in orchards helps farmers make improved decisions regarding farm management. Research in this field has ranged from orchard mapping, autonomous driving for farm vehicles, and segmentation and classification of the farm.

Work done by Moorehead, Wellington, Gilmore, and Vallespi (2012) and Subramanian, Burks, and Arroyo (2006) examines autonomous vehicle guidance in orchards. Using a combination of visual and laser sensors to detect trees and drivable terrain, along with GPS for localization, they built a map of the environment. Their developments led to autonomous spraying operations across the farm, however they did not aim to detect or map individual trees but rather observed the presence of trees for optimized sprayer control.

Ultrasonic, image, and laser sensors have been used by Wei and Salyarii (2005) and Tumbo, Salyani, Whitney, Wheaton, and Miller (2002) for predicting canopy volume and height for individual trees. From these data, tree health and yield could be inferred and associated with trees that have been manually labeled. We wish to automate individual tree identification to help build a tree inventory to support a database of crop health, yield, and other tree-specific information. One solution could involve tagging each tree in the farm with a unique RFID or QR code, which a ground vehicle could recognize accurately. However, we seek a modular and cost-/labor-efficient detection solution, where minimal additional infrastructure is needed, allowing us to easily deploy the system on different farms.

Geometry- and feature-based model fitting have been popular choices in outdoor scenes for remote/unobtrusive detection of individual trees. Using data from lasers on a ground vehicle, Nielsen, Slaughter, Giever, & Upadhyaya (2012) fit Gaussian mixture models (GMMs) onto a row of trees. A single Gaussian cluster represents a tree (assuming they are well separated in the point cloud data), and the ideal number of clusters (representing the count and position of individual trees) is evaluated through an information criterion algorithm. With regard to feature-based approaches, tree detection can be performed by classifying points using shape descriptors and shape functions (Lalonde, Vandapel, Huber, & Hebert, 2006; Pauling, Bosse, & Zlot, 2009). Airborne laser scanning has also been used for tree detection. In Yu, Hyypä, Kaartinen, and Maltamo (2004), trees are segmented based on height variations captured by a downward-facing laser sensor.

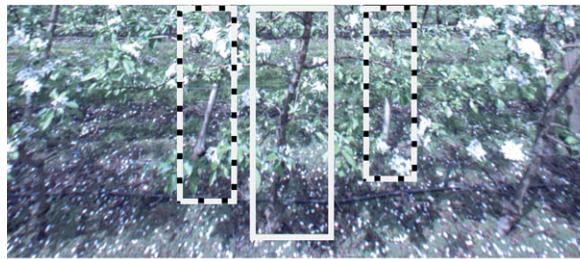
Autonomous operations requiring intrarow navigation have previously benefited from individual trunk detection. In Hamner, Singh, and Bergerman (2010), a 2D lidar configured in the horizontal plane at trunk height is used to detect laser returns from the trunks (and other objects) to model the tree rows. The row configuration can then be used to plan a traversable path down the center. The coarse nature of the trunk detection (with a large number of false returns) is sufficient for path planning, but it does not act as a robust mechanism for trunk detection. A more robust particle-filter-based approach is proposed in Zhang et al. (2013), where individual trunk point clouds are modeled as Gaussian distributions and detected one by one as a ground vehicle traverses down the row.

The methods mentioned above have been designed primarily for trees planted in a traditional squat formation. The configuration of a trellis structured orchard, on the other hand, adds additional challenges to tree detection. The heavily intertwined and overlapping trees blur the definition of an individual tree, preventing the use of canopy geometry or feature-based models for segmentation. Trellis trees often have thinner trunks with increased low-hanging foliage, compared to their free-standing counterparts. This results in a reduced signal-to-noise ratio in lidar data, which

creates a challenge for any method that relies upon local detection (such as the ones mentioned above) of trunks within individual lidar frames. Furthermore, these methods provide detection solutions in real time as they are directed toward farm mapping and vehicle navigation. As we are motivated by an information management system that spans the entire farm, real-time operation is not a necessity. With postprocessing, we can obtain more accurate detection results as we can consider all the data from the entire row at once, rather than sequentially on-the-fly. Wellington and Campoy (2012) extract tree segments in this way by considering entire rows, and by leveraging from the regular distribution they are able to obtain accurate detection. Jagbrant, Underwood, Nieto, and Sukkarieh (2014) compared offline and online solutions to tree segmentation, showing that improved accuracy can be obtained using all the data offline, if real-time operation is not required.

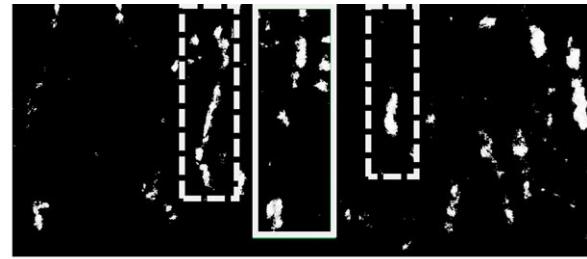
In Wellington and Campoy (2012), a tree segmentation method is presented that uses laser data on a citrus orchard. By splitting the point cloud data into thin slices along an orchard row, slice by slice classification is performed to separate trees, tree boundaries, and gaps between trees. A hidden semi-Markov model (HSMM) predicts the most optimal state sequence giving rise to the set of quantifiable observations. Adjacent trees are differentiated from each other by considering the changes in heights of the slices of point cloud data. This method is able to encapsulate orchard structure by setting state duration probabilities, which explicitly encode the repetitive tree spacing as a constraint. Individually segmented trees form a tree inventory over the farm where each item is localized using GPS, resulting in a complete geographic information system (GIS) for orchard management. However, when operating through orchard rows, the tall canopies can often hinder the GPS satellite reception, reducing the reliability of the system. One possible solution is provided by Jagbrant et al. (2014), who instead store a topological tree inventory rather than a metric one. Tree localization is then performed by matching sequences of unique descriptors from different scans.

As stated before, in a trellis structured orchard with heavily intertwined trees, the concept of tree segmentation is ambiguous. Additionally, an observational model based on tree height changes is not representative of individual tree locations. Instead, as the trees in orchards are often pruned at the bottom, the trunks are generally visible and act as distinct markers for the individual trees. The particle filtering approach in Zhang et al. (2013) or a simple Bayes filtering coupled with a vehicle motion model could be applied here but would not be able to provide the globally optimal results that can be obtained with an HSMM. Previous work (Bargoti et al., 2014) presented a perception pipeline to detect trunks by incorporating an HSMM framework like the one used in Wellington and Campoy (2012) and Jagbrant et al. (2014). A topological map of the individual trees was built in an orchard hosting



(a) Image from a camera on-board the mobile platform.

Figure 2. Apple tree trunks captured using the camera onboard the research vehicle. The trunk in the solid frame is from the foreground row. The dashed frames contain tree trunks from the adjacent background row that appear in the FOV of the camera. Without depth information, a classifier cannot tell the two apart.



(b) Pixel-wise classification results for trunk detection.

the tightly meshed V-trellis structure. The method was tested over a half-hectare plantation and produced trunk detection results with 89% accuracy. A limiting factor in the pipeline was a suboptimal observation model based on line detection in lidar data to gather tree trunk candidates. It was observed that during operation in areas with heavy foliage and/or tall grass, tree trunks and overhanging foliage were not discernible within the lidar observation model. As a result, false-positive returns were observed, where nontrunks were classified as trunks.

Image-only classification approaches such as those presented in Dey, Mummert, and Sukthankar (2012), Jimenez, Ceres, Pons (2000), and Nuske et al. (2014) have been successful in orchard environments. Additionally, the task of scene parsing has been approached with a wide variety of methods such as the ones presented by Farabet, Couprie, Najman, and Lecun (2012) and Munoz, Bagnell, and Hebert (2010). One can adopt such techniques to semantically label the tree trunks on images captured sequentially along the length of a row. Accurate scene parsing can eliminate the ambiguity between overhanging foliage, tall grass, and tree trunks. However, when working in natural environments, the variable lighting conditions and the inability to capture sufficient variety within training examples limit the scene parsing performance (Hung, Nieto, Taylor, Underwood, & Sukkarieh, 2013). More importantly, in a trellis structured apple orchard, trees from background trellis rows appear in images containing the foreground trees. An example is shown in Figure 2(a), where trunks within the dashed frames that are from the background row appear at a similar position in the image as the foreground tree in the solid frame. A classification algorithm trained to detect tree trunks is unable to tell apart the foreground trees from the background [Figure 2(b)]. Stereo vision would provide relevant depth information to filter these data and also provide a spatial representation so that we can add a regular tree-spacing constraint. However, empirically we have observed that the complex natural foliage structure can cause

difficulties for depth estimation using current stereo algorithms.

This paper presents an extension to the framework introduced in Bargoti et al. (2014) with the inclusion of image data in the trunk detection process. To utilize the strength of the two modalities, we build upon the observation model previously generated by lidar data alone by performing image classification for trunk detection. This lowers the trunk/nontrunk ambiguity mentioned above, reducing the total number of false-positive returns. Additionally, where the previous work segmented a single V-trellis type during a particular time of season (preharvest), this paper reports on the performance of the automated trunk detection pipeline over different trellis formations (both I and V structures), apple varieties (Pink Lady and Granny Smith), and at different times of the year (flowering and preharvest seasons).

3. TRUNK DETECTION PIPELINE

The trunk detection process is carried out by a pipeline that accepts raw lidar and image data as input and provides a metric and topological representation of individual trees over the farm. This process is summarized in Figure 3 and described in full in this section.

The primary input into the pipeline is lidar data captured over a block at the apple farm. A block is defined as a set of adjacent trellis rows consisting of the same type of trellis structure. The lidar data are in the form of a georeferenced point cloud, which is a set of raw measurements that represent the geometry of the orchard block [top view shown in Figure 3(a)]. Within the lidar sensor frame, a fixed distance threshold is applied to ensure that over a single row traverse, data from adjacent rows (seen through gaps in the foliage) are discarded. Individual rows are then automatically segmented according to the vehicle heading angle along each row.

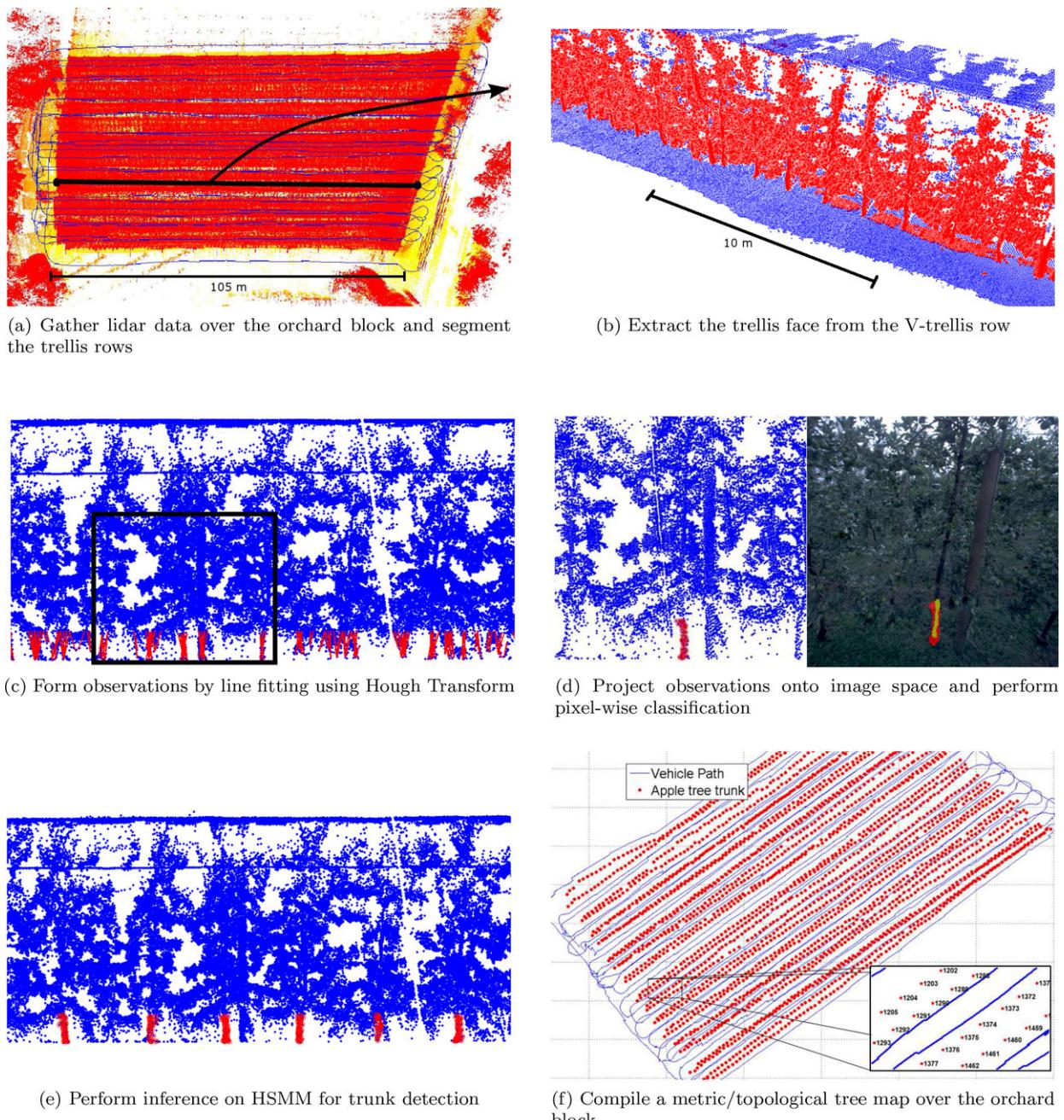


Figure 3. Trunk detection pipeline (a) → (f). (a) Point cloud representation of the orchard block. Points are colored by elevation, and the blue path represents the vehicle trajectory. Lidar data from a single row are extracted by applying a vehicle heading threshold. (b) The front trellis face (shown in red) is segmented out of the single V-trellis row. (c) Hough transform applied to the trellis face yields possible trunk candidates. (d) Trunk candidates are projected into the image space (red pixels overlaid on the image), and pixelwise image classification is performed to detect trunks (pixels classified as trunks are shown in green). The number of pixels classified as a trunk is fused with the lidar observation to form a combined trunk candidate likelihood. (e) Robust trunk detection (points in red) is obtained by performing inference on the HSMM. (f) Repeating this over all the rows, a GPS localized tree trunk map is formed over the orchard block. Vehicle path in blue, trunks in red. The inset shows the individually labeled trunks.

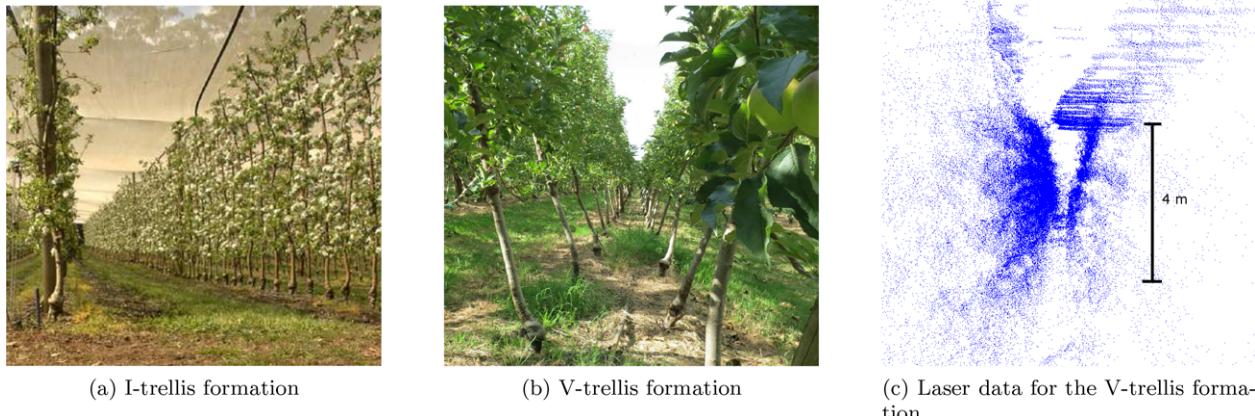


Figure 4. Two different trellis formations at the apple orchard. Part (a) is a side view of an I-trellis plantation. In (b) we are looking through the center of one V-trellis, with (c) displaying the equivalent point cloud. The scan was acquired from the left-hand side. The density on the right is lower due to occlusion.

The two trellis structures evaluated in this paper are the standard I-trellis and the Güttingen V-trellis (Figure 4). A point cloud representation of the latter is shown in Figures 4(c) and 3(b). The I-trellis structure is comprised of a vertical support system hosting apple trees planted evenly through the row. The V-trellis structure, on the other hand, consists of two closely planted halves, hereby referred to as trellis faces. The I-trellis and V-trellis trees are planted on average 0.75 and 1.5 m apart, respectively. The rear face in the V-trellis has a lower lidar point cloud density due to occlusions, and therefore trunk segmentation needs to be performed on the front trellis face [nearest to the vehicle, as shown in Figure 1(a)].

On the trellis face, we capture observations that relate to states representing trunks and the gaps between the trunks. These observations are then updated by incorporating image data that are captured at the corresponding locations. By discretizing the multimodal observations into vertical slices along the row, the HSMM provides a probabilistic framework for estimating the most probable state sequence resulting in the set of observations. A parallel sequential detection process can also be executed to estimate support pole locations, as demonstrated in Bargoti et al. (2014).

3.1. Trellis Segmentation

Figures 4(b) and 4(c) show photographs from a hand-held camera, and the lidar view from inside the V-trellis structure. The trunks from the trellis faces that form the V structure are separated by roughly 0.5 m at the ground. We aim to segment the two halves of the point cloud shown in Figure 4(c). Ground removal causes the faces to be more easily spatially separable, and it also causes the trunks to appear as more distinct linear structures in the lidar points clouds.

3.1.1. Ground Removal

Ground removal in laser data has been a subject of investigation in many mapping and segmentation problems (Douillard et al., 2011). However, given the constraints on the geometry of the orchard, and the vehicle and sensor configuration, a local height threshold was found to be sufficient. To simultaneously account for nonuniform terrain and GPS altitude errors due to poor satellite visibility, we artificially set the altitude coordinate of our localization estimate to zero everywhere, prior to georeferencing the laser data. There can still be variations in the ground height between the tree trunks and the ground where the vehicle drives, therefore an adaptive threshold is evaluated over sliding windows down the length of the row. For a given window, the point cloud data are binned into discrete divisions along the vertical axis. The large number of laser returns from the ground results in a spike in this histogram near the ground level, as illustrated in Figure 5. The height threshold for ground subtraction is determined by the edges of the spike (shaded region in Figure 5), which can be evaluated by extracting the gradient of the histogram. This threshold is linearly interpolated over the windows along the length of the row and then applied to the entire point cloud for ground removal. A similar approach is used to remove data from the netting cover that forms a ceiling above the orchard. In our implementation, the sliding window was configured to cover approximately three to six trees with a bin size of 0.05 cm, but the performance was not found to be sensitive to this parameter.

3.1.2. Face Segmentation

To separate the two faces within the V-shaped structure, we fit a piecewise linear boundary between them, all the way along the row. A single line is less robust, because

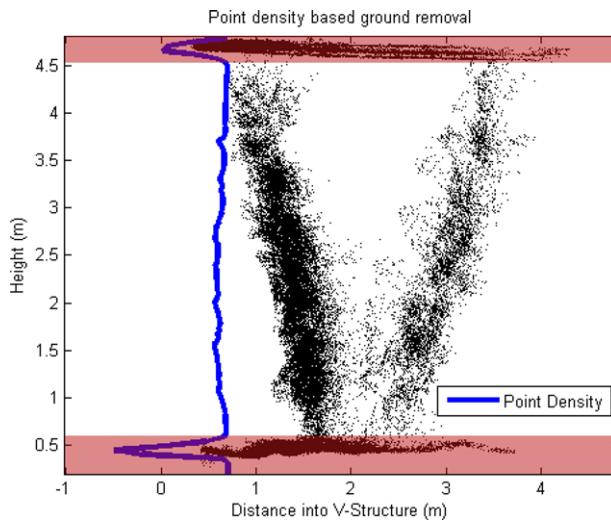


Figure 5. Side view of a point cloud representation of the V-trellis structure in one sliding window down the row. A histogram of point density is evaluated along the height of the trellis to detect the ground and top cover location, resulting in a local height threshold.

deviations occur both in the physical row and in the data due to GPS errors. The boundary is located by observing changes in the lidar point density over sliding windows along the row, as illustrated in a top view representation of the row in Figure 6. The sliding window is configured at roughly three to six trees with no overlap. If the window is too narrow, noise from the point cloud dominates, and if the window is too large, local deviations in the boundary are missed. The point density profile for a single window is illustrated in Figure 7. It contains two peaks representing the front and rear face (smaller second peak due to lidar occlusions). A Gaussian mixture model (GMM) with two modes is fit to this, and the means represent the central position of the two faces. The boundary point between the faces is taken as the midpoint between the means (illustrated in the figure). Additionally, the GMM parameters from the preceding window are used as priors for the next window, which allows us to filter out peaks from excessive foliage or inaccurate ground removal. Joining together the boundary points along the row results in the boundary line illustrated in Figure 6. The points to one side, which represent the trellis face closest to the vehicle, are shown in red in Figure 3(b).

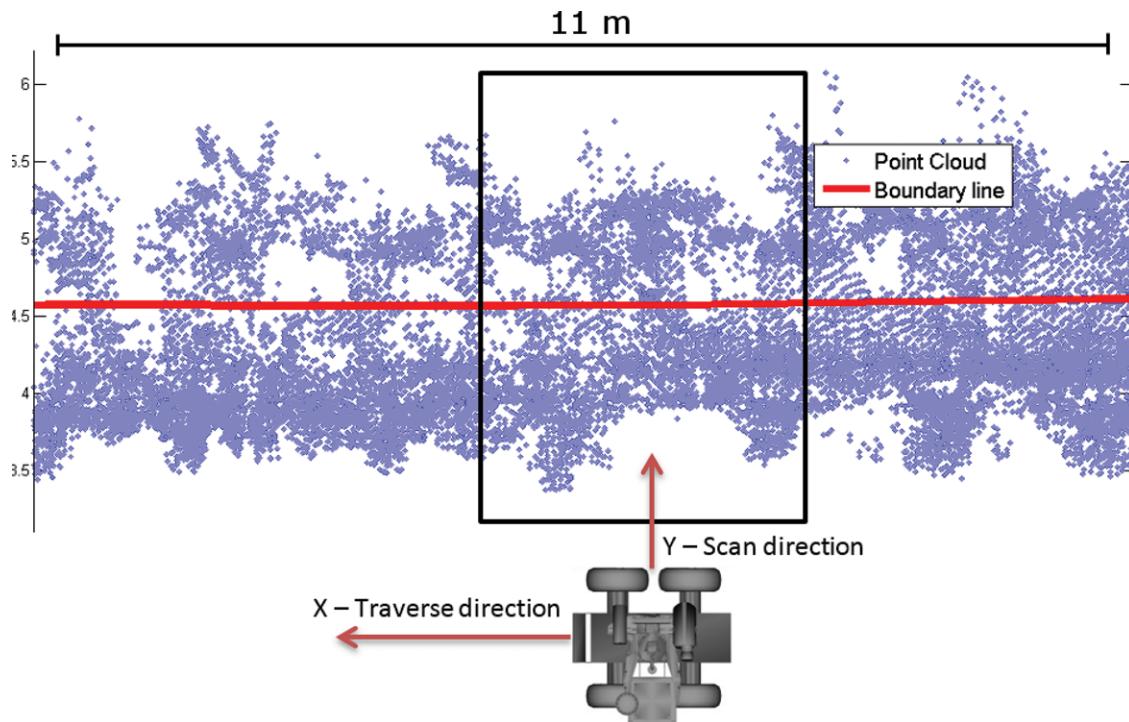


Figure 6. Top view of the point cloud. The dataset was obtained from the perspective shown in the image resulting in a higher point density on the closer trellis face. Discrete boundary points are found at sliding windows along the x direction (black rectangle). The resultant separation boundary is illustrated by the solid line between the trellis faces.

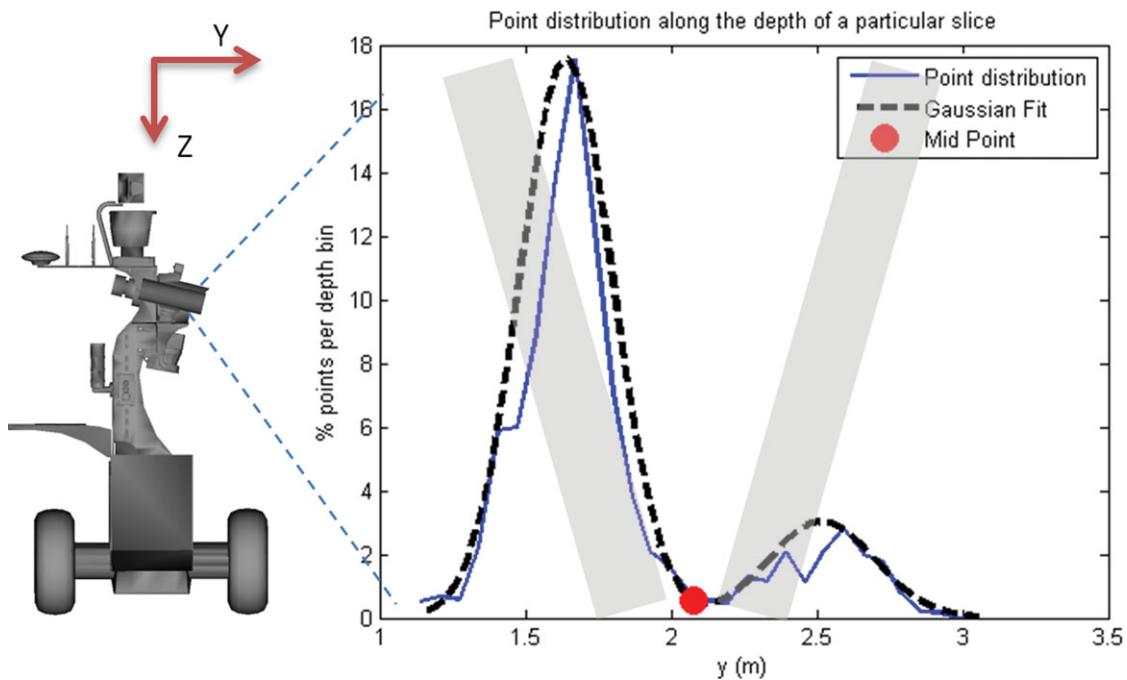


Figure 7. Distribution of points in a sliced segment of the row as observed over data captured by the ground vehicle lidar. The gray patches represent the two faces of the V-trellis. A GMM of two modes is fit to the data (dashed line), and the midpoint between their means is the boundary point between the faces for the given slice.

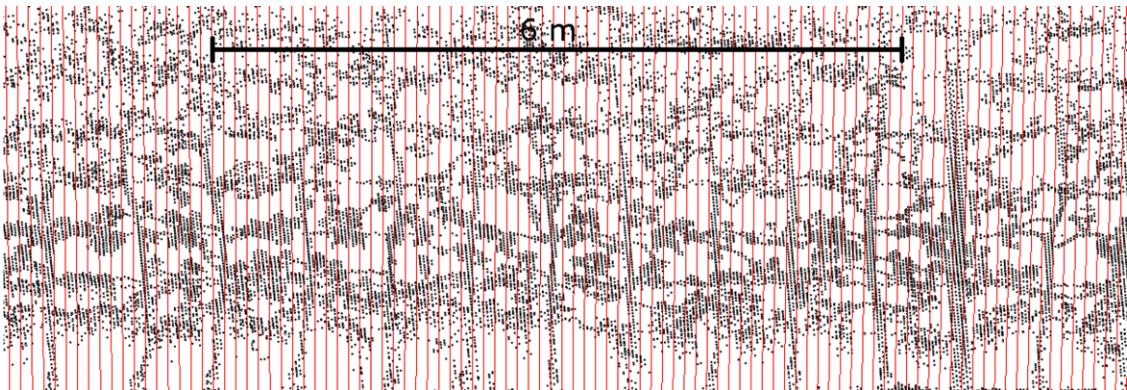


Figure 8. The 2D point cloud representation of the I-trellis face is discretized into slices along the direction of the row. Trunk detection involves identifying all slices that contain a trunk.

3.2. Hidden Semi-Markov Model

The second half of this pipeline focuses on the detection of individual tree trunks. For this we use a hidden semi-Markov model (HSMM) (Bilmes et al., 1998). HSMMs are used to estimate the most likely sequence of states given a sequence of sensory observations. Operating offline on the complete data, they are able to produce results that are globally smooth. They have been used in diverse areas, such as

speech recognition, human activity recognition/prediction, and handwriting recognition (Yu, 2010). To formulate the trunk detection problem as a HSMM, we discretize the point cloud representation of the trellis face into slices (roughly the trunk width) along the length of the row, as illustrated in Figure 8. The slices are assigned states corresponding to whether they represent a trunk or a gap between trunks. The models for the sensory observations relate to the likelihood of having a particular state for some observed data. A

lidar observation model was used previously (Bargoti et al., 2014), where lines were fit to the lidar data, and longer lines meant higher likelihood of a trunk. In this paper, we provide an extension to this observation model by incorporating image data, whereby pixelwise classification of images captured simultaneously with the lidar is used to refine the probabilities of the presence of a trunk.

3.2.1. Lidar Observation Model

The purpose of this observation model is to capture the differences between a tree trunk and the gaps between trunks within the lidar point cloud data. For this, we focus on the geometric properties of tree trunks that set them apart from their surroundings.

There are several approaches in the literature that are used to estimate the linearity or cylindrical nature of regions of point clouds. In Zhang et al. (2013), the trunks of trees in a traditional squat formation are represented as ellipsoidal structures with Gaussian distributions. Similarly, in Vandapel, Huber, Kapuria, and Hebert (2004), principal component analysis (PCA) was used to define the point cloud shapes as either linear, planar, or scattered in order to classify them. In our initial experiments, GMMs were trained using hand-labeled classes such as ground, trunks, and foliage; however, data obtained by sampling trellis structured orchards using a mobile platform are typically sparse. In trellis orchards, where the trunks are thin (~ 5 cm), combined with mobile scanning at adequate velocity to cover large areas of the farm, the point cloud density is too low to extract discriminative shape information.

In another approach that relies on 3D point statistics, minimum spanning trees are used to grow regions of data approximated with ellipsoidal structures (Pauling et al., 2009). This works by connecting neighboring points according to an edge weight equivalent to the similarity in the PCA feature space. Empirically, once again due to a sparse representation of the trunks, it was a difficult task to distinguish foliage from tree trunks.

We chose instead a direct line fitting approach using the Hough transform (Illingworth & Kittler, 1988) on a region of interest governed by the maximum length of a trunk from the ground to the foliage. Given a parametric equation of a line, a voting procedure is used in the parameter space to find lines of best fit on the input data. For points in 3D space (as in the case here), a 4D Hough space is needed for line detection, which is computationally expensive (Bhattacharya, Liu, Rosenfeld, & Thompson, 2000). An alternative is to detect edges using intersections between fitted planes in 3D space. This works well in indoor environments where objects have flat surfaces and sharp edges, but it is infeasible in an outdoor scene where planar surfaces are infrequent. Instead, we leverage from the inherent 2D structure of the trellis and flatten the 3D data onto a 2D plane, and we apply a 2D Hough transformation

over data that spans the entire length of the row.¹ The trunks typically grow approximately vertically with respect to gravity, regardless of the underlying terrain slope. We restrict the line gradient to $-15^\circ \leq \theta \leq 15^\circ$ to account for variations in trunk growth against the gravity vector.

In the raw form, the local line detection results in a large number of false positives. A simple Bayes filter could be applied here to reduce the number of false returns, however the global filtering and smoothing nature of the HSMM is more suitable for this operation as it takes all the data into consideration. We distribute the fitted lines into the state slices illustrated in Figure 8. The length of the fitted line in a given slice relates to the likelihood of the slice representing a trunk or a nontrunk. A 9.5-m-long subsection of a row from the orchard was chosen to demonstrate this and the proceeding components of the pipeline. In Figure 9(a), the point cloud representation of this section is overlaid with the maximum line length per slice (normalized to the longest line and rescaled for visualization). The observation likelihood function is presented in Section 3.2.3. This approach is able to detect all valid tree trunks but contains false positives when there is excess overhanging foliage (as seen in some parts of the figure). The resulting observation model has a high recall but a low precision rate, which leads to segmentation errors within the HSMM framework (Bargoti et al., 2014). The next section examines augmenting this observation model by incorporating image data.

3.2.2. Image Observation Model

The lidar trunk observation model is updated by using image classification techniques. The low precision, high recall line-fit results from lidar data [Figure 3(c)] help to narrow down the search region within the image space, where we improve precision by detecting false-positive occurrences. First, we cluster the lidar points corresponding to each detected line. A 1D K-nearest-neighbors approach is used to group points together, which we denote as lidar trunk candidates. When compared to the true trunk locations, these candidates have low precision due to returns from foliage, but near perfect recall as lidar always returns from a trunk.

We project the lidar trunk candidates onto the image space (Underwood, Scheding, & Ramos, 2007), obtaining the corresponding visual information from the camera in the form of image pixels. These pixels are generally sparse due to the sparsity of the lidar data, so they are dilated to create a smooth area representing the trunk candidate in the image [examples overlaid on the image in Figure 3(d) and in Figure 10]. Standard image classification techniques can then be applied to these collections of pixels. For pixelwise classification, we use a sparse autoencoder with a soft

¹RANSAC line fitting was also tested using a sliding window approach, but the Hough transform is more efficient and conceptually simple as it processes the entire row at once.

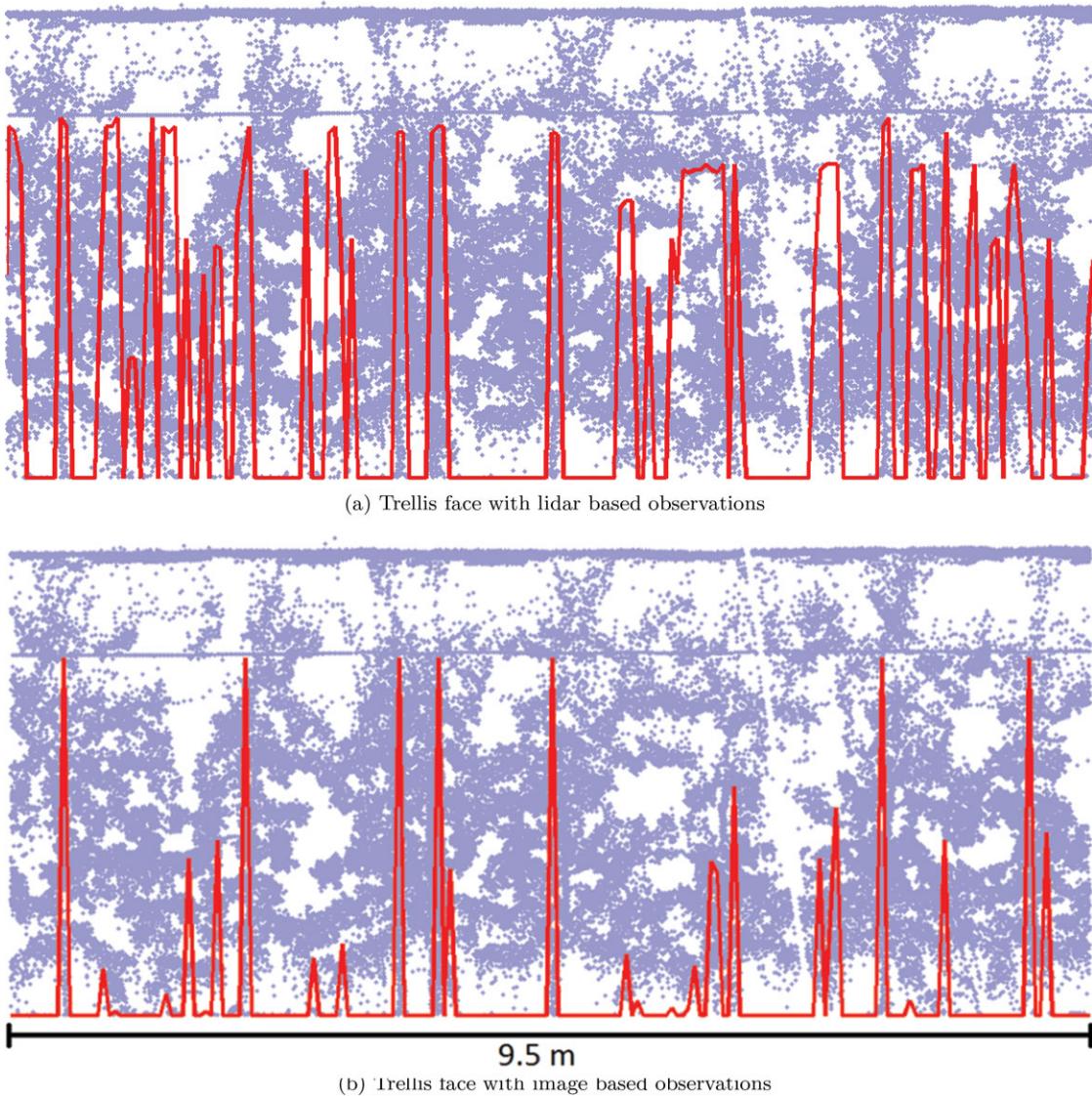


Figure 9. Trunk observations derived from (a) lidar and (b) combined lidar/imagery overlaid on the trellis face (raw lidar point cloud in the background). The observations have been rescaled for visual interpretation. The lidar-based observations generated through line fitting contain a large number of false-positive returns. Image-based observations generated through pixel classification have much higher precision.

max function, due to the success of this method in other orchard classification applications (Hung et al., 2013).² The classification model is built by training a two-class classifier for differentiating trunks and nontrunks. The training set is randomly sampled from the entire dataset so as to capture the variable lighting conditions at different times of the day.

²The aim of this work was not to search for the best image classification algorithm, but one that is able to perform pixelwise classification with sufficient accuracy.

Figure 10 shows the classification result from two scenarios, where a true positive lidar trunk candidate is confirmed as a trunk in the image space [Figure 10(a)], and a false-positive lidar trunk candidate is rejected as a trunk in the image space [Figure 10(b)].

For use within the HSMM framework, the image observations [overlaid on the pointcloud in Figure 9(b)] are defined to be the total number of pixels classified as trunks that match the lidar trunk candidates' locations in the image. A zero trunk pixel count is registered for slices that

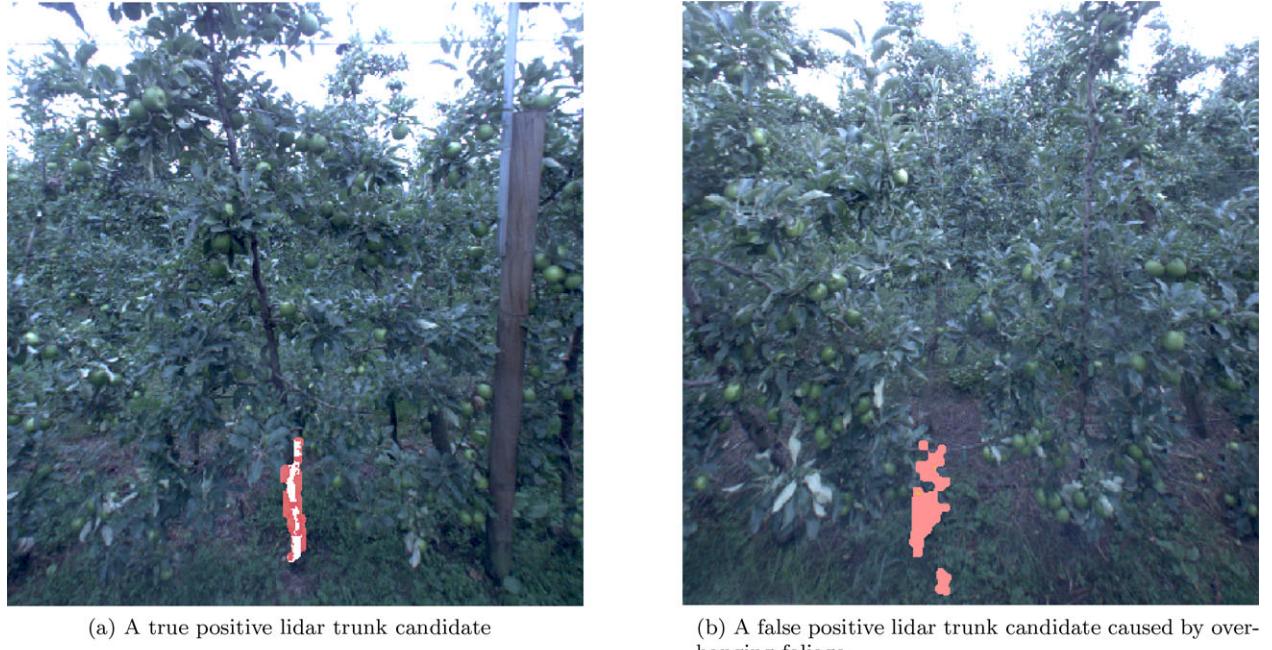


Figure 10. Lidar trunk candidates projected onto the image space. Using pixelwise image classification, in (a) the lidar trunk candidate is classified as a trunk (white overlay), whereas in (b) the incorrectly identified lidar trunk candidate is classified as a nontrunk (darker overlay).

do not contain any lidar trunk candidates. Compared with the lidar-based observations, the false-positive rate has decreased significantly. We can now add structural constraints in the form of regular tree spacing along the orchard row and acquire the trunk locations for entire rows through the use of the HSMM.

3.2.3. Model Parameters

The states in the HSMM are defined as $S = \{S_{\text{trunk}}, S_{\text{gap}}, S_{\text{row-end}}\}$. With a perfect inference, a trunk slice state indicates the presence of a trunk, and a group of gap states will represent the area between two trunks. A row-end state is also introduced to model the empty area before the first tree trunk and after the last. The size of this area is arbitrary as it is dependent on how the dataset was acquired and how the individual rows were segmented. A state at slice n is then denoted as q_n , where $n \in [1 : N]$ for a total of N slices.

For a sequence of observations $\mathbf{O} = \{O_1, \dots, O_N\}$, the HSMM aims to find the state sequence $\mathbf{Q} = \{q_1, \dots, q_N\}$ that best represents the data:

$$\arg \max_Q P(\mathbf{O}|\mathbf{Q}, \pi, A, B, C). \quad (1)$$

Here π is the initial state distribution, the probability of a state in the first slice. In the set configuration, this will be

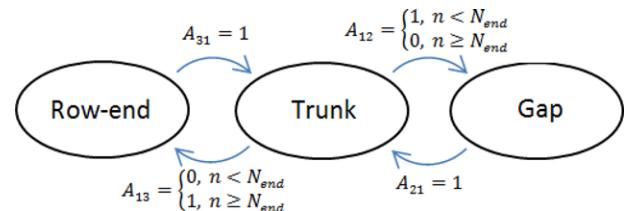


Figure 11. Transition probabilities between states in the HSMM.

the row-end state,

$$\pi = P(q_1 = S_i) = [0 \ 0 \ 1] \ i \in \{\text{trunk, gap, row-end}\}. \quad (2)$$

The state transition matrix A represents the probability of moving between states as we go from slice to slice. Its elements are

$$a_{ij} = P[q_{n+1} = S_j | q_n = S_i], \ i, j \in \{\text{trunk, gap, row-end}\}. \quad (3)$$

The model transition probabilities are graphically illustrated in Figure 11. The most likely transitions are between the trunk and gap states. Naturally, it is only possible to transition to/from the row-end state at the end of the rows,

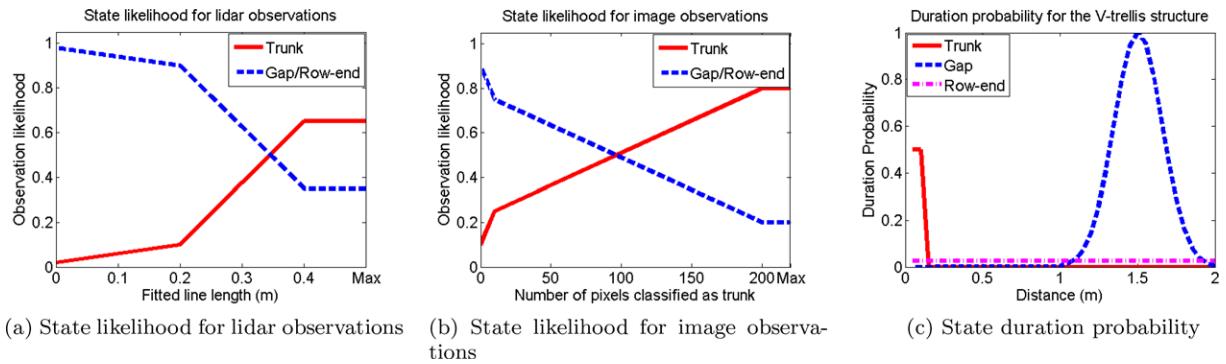


Figure 12. Observation and duration probabilities for the HSMM.

and it is modeled with transition probabilities that are a function of slice position. Starting off with the row-end state [Eq. (2)], the only transition possible is to the trunk state. For the length of the row, a trunk state is modeled to go to the gap state except when near the end of the row, N_{end} (roughly equal to the gap width between two trunks).

The emission probability B states how likely an observation will occur for a given state,

$$B_i = P(O_n | q_n = S_i) \quad i \in \{\text{trunk, gap, row-end}\}. \quad (4)$$

For lidar-based observations [Figure 9(a)], a longer line fit equates to a greater likelihood of a trunk state, and due to its low recall rate, the absence of a line fit will most likely represent the gap state. The manually tuned emission probability is shown by the solid line in Figure 12(a), the inverse being true for the gap/row-end state (dashed line). For image-based observations [Figure 9(b)], a larger number of pixels classified as trunks is indicative of the presence of a trunk state, however a low trunk pixel count does not guarantee a gap state due to instances of pixel misclassification or trunk occlusion by foliage. This reasoning gives rise to the probability distribution shown in Figure 12(b). For both the lidar and image data, the observation models are empirically validated in Section 4.1 using hand-labeled ground truth data. A multimodal observation model then combines both of these emission probabilities,

$$B = P(O|q = S_i) = P(O_L|q = S_i) \times P(O_I|q = S_i) \quad i \in \{\text{trunk, gap, row-end}\}, \quad (5)$$

where O_L is the lidar-based observation and O_I is the image-based observation for a given slice.

Finally, the duration probability C encapsulates the structural order of the orchard. With slices of $w_s = 5$ cm width, we expect a gap state to last as long as the mean separation between trunks in the farm. Within the HSMM

framework,³ we can define a state duration probability:

$$C_i(d) = P(S_i \text{ lasts for } d \text{ observations}). \quad (6)$$

Distributions for the different states are shown in Figure 12(c). The gap duration has been modeled as a Gaussian distribution by manually estimating the average trunk separation at the orchard block. The trunk states have a duration of one to two slices (representative of the average trunk width), and the row-end states can last for an arbitrary duration (to account for the variability in whole-row segmentation).

We run the Viterbi inference algorithm (Bilmes et al., 1998) to search for an optimal solution for Eq. (1). To illustrate the process, Figure 13 shows the results of the inference algorithm on the sample data presented in Figure 10. The state estimates produced by using lidar data only are illustrated by the dashed lines, and estimates produced through a combination of lidar and image data are shown with a solid line. The combined observation model is able to correctly classify the tree trunks. The trunk classification results can be projected back to the original point cloud, as shown in Figure 3(e). Secondly, through this labeling, we can reevaluate the average trunk separation for the given row, obtaining a more accurate and row-specific gap duration model illustrated in Figure 12(c). By running inference once more with the updated parameters, we are able to improve upon the trunk detection performance.

4. TRUNK DETECTION RESULTS

The trunk detection pipeline was tested at an apple orchard near Melbourne, Australia (Figure 1). The testing platform “Shrimp” [Figure 1(a)] is a perception research ground vehicle, built at the Australian Centre for Field Robotics at The University of Sydney. It is equipped with a vertically oriented SICK LMS-291 2D lidar directed perpendicular to

³Self-state transitions experience an exponential decay when working with a HMM. This ability to explicitly set state durations is what differentiates an HSMM from an HMM.

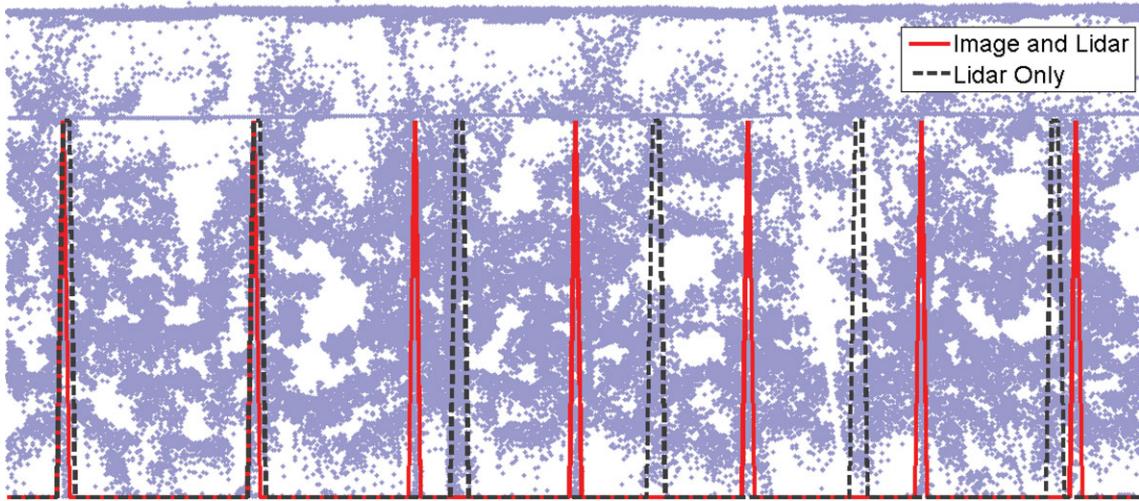


Figure 13. Trunk and gap estimates obtained by running inference on the HSMM, 1:trunk, 0:gap. Inference performed using two different observation models; results from lidar only shown by the dashed line (containing misclassified trunks) and results from image and lidar combined shown by the solid line (correct trunk classification).

the vehicle's direction of travel. The sensor captures range data at 75 Hz as it sweeps along the orchard rows. On top of the vehicle, there is a spherical imaging camera from which we can extract image data (captured at 5 Hz) that overlap the field of view of the lidar. By using a tree-line following algorithm, the vehicle traversed along the rows of different orchard blocks collecting raw lidar [shown in Figure 3(a)] and image data.⁴

To assess the performance of the pipeline, a variety of block types were scanned:

- Two different trellis structures: the vertical I-trellis [Figure 4(a)] and the Göttingen V-trellis [Figure 4(b)]. The V-trellis structures require the extra face segmentation procedure within the pipeline, while the I-trellis does not. Additionally, the apple trees are planted closer together in the I-trellis configuration (~ 0.75 m apart) than on each face of the V-trellis system (~ 1.5 m apart).
- Different times of the year: data were gathered during flowering in October [Figure 14(a)] and just prior to harvest in March [Figure 14(b)]. During the flowering season, the trees are relatively bare with minimal foliage, whereas during preharvest thicker foliage causes adjacent trees to mesh and blend more significantly.
- Different apple varieties: Pink Lady and Granny Smith. The Pink Lady apples were planted on a block hosting the V-trellis structure. The dataset was captured prior to harvest only, and it was used for lidar-only trunk detec-

tion in Bargoti et al. (2014). The Granny Smith apples were planted on both V- and I-trellis types, which were scanned both during flowering and preharvest.

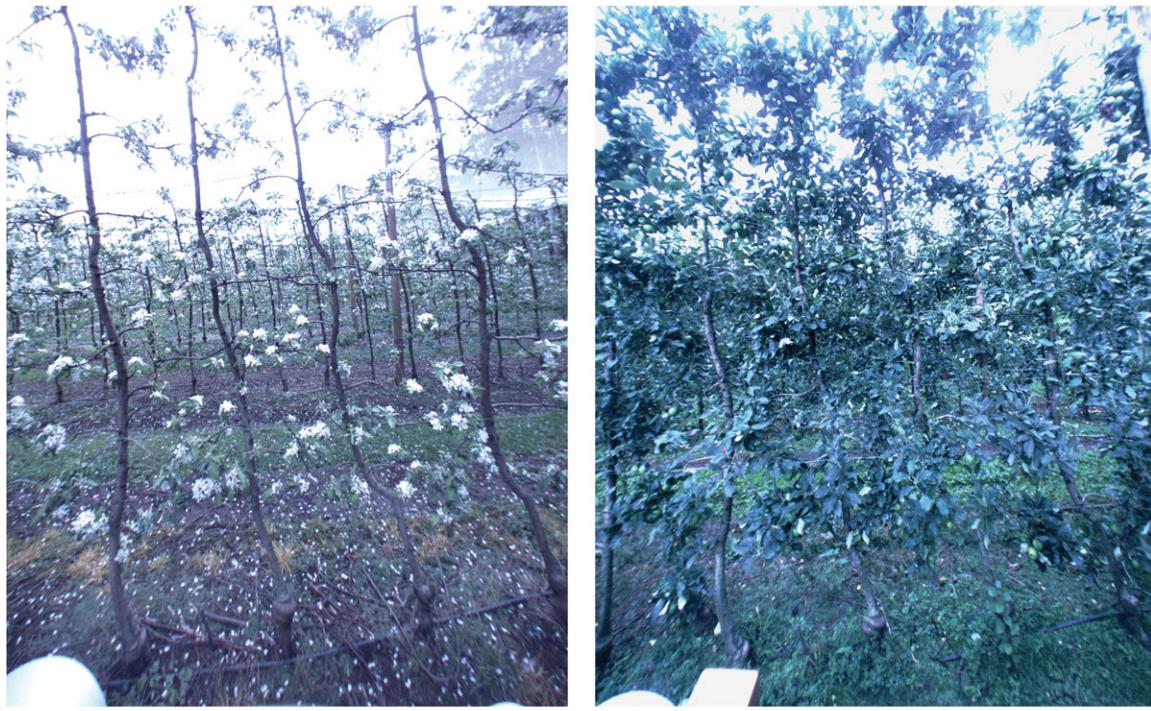
- Different vehicle speeds: ~ 0.5 and ~ 1.5 m/s. A faster traversal means sparser lidar points, resulting in a weaker representation of the already thin trunk structures. Due to successful trunk detection results observed in Bargoti et al. (2014) from lidar data collected while traversing at faster speeds, we scanned only two rows from each trellis type at the slower speed, for comparison purposes.

In total, the vehicle scanned three blocks covering a total area of 1 ha. With field trials conducted over different times of the year, the data collected covered a total of 1.6 scanned hectares. This included vehicle traversal over 41 rows, a vehicle trajectory length of 7,655 m, scanning 6,133 trees in total.

An onboard Novatel Global Positioning Inertial Navigation System (GPS/INS) provided estimates of the vehicle trajectory, which was used to produce a georeferenced point cloud representation for each block. User input was required to specify the trellis structure type, to estimate the average tree trunk separation, and to provide the image classification algorithm with training data. Training on 10 trunks labeled on randomly selected images over each block provided sufficient accuracy for the pixelwise classification task.

The trunk detection results for each block are listed in Table I. We compare results obtained by using the previously proposed lidar-only observation model (Bargoti et al., 2014) and the new image- and lidar-based

⁴A mix of remote-control and automated driving was used to gather different datasets. In all cases, the robot drove along the centerline of the orchard rows.



(a) Apple trees during flowering season

(b) Apple trees during harvest season

Figure 14. Apple trees on an I-trellis structure during flowering and the preharvest season. In the latter there is more overhanging foliage next to the tree trunks.

Table I. Apple orchard trunk detection results using image and lidar data captured at 1.5 m/s.

Block Type\Observation Type				Lidar			Image and Lidar		
Dataset	Trellis	Fruit	# Trees	Count ¹	TP	Acc (%)	Count	TP	Acc (%)
1. Preharvest	V	PL ²	2494	2,472	2,238	89.7	2,462	2,389	95.8
2. Preharvest	V	GS ³	939	870	442	47.1	918	815	86.8
3. Preharvest	I	GS	672	583	323	48.1	667	628	93.5
4. Flowering	V	GS	939	938	920	98.0	939	930	99.0
5. Flowering	I	GS	659	660	648	98.3	660	652	98.9

¹Estimated tree count, True Positives (TP) + False Positives.

²Pink Lady.

³Granny Smith.

observation model from Section 3. Table II shows the trunk detection results on the denser lidar data captured at lower velocity.

Ground truth trunk locations were obtained by manually selecting points that correspond to tree trunks on the point cloud data. For data in areas of thick foliage, where it was difficult to distinguish the tree trunks using manual, visual inspection of only the point cloud data [e.g., areas of Figure 16(b)], we analyzed the corresponding image data to

confirm the presence of a tree trunk. The performance of the trunk detection pipeline was evaluated by comparing the state estimation results (Figure 13) with the ground truth. A true positive trunk detection refers to a trunk state estimated within 5 cm of the ground truth trunk location (half the width of a trunk). A false positive refers to trunk state estimates outside this threshold. The trunk count estimate is a sum of true positives and false positives. False negatives occur when actual trunks are labeled as gap states, and they are

Table II. Apple orchard trunk detection results using image and lidar data captured at 0.5 m/s.

Block Type\Observation Type				Lidar			Image and Lidar		
Dataset	Trellis	Fruit	# Trees	Count	TP	Acc (%)	Count	TP	Acc (%)
6. Flowering	V	GS	162	155	136	80.1	161	158	97.5
7. Flowering	I	GS	268	268	266	99.3	268	266	99.3

the difference between the actual trunk count and the true positive trunk count. True negatives were not counted as the area between trunks is comprised of multiple consecutive gap states, the number of which depends on the resolution of the slice width. The pipeline performance/accuracy was determined as a ratio between true-positive trunk detection and the total number of trees in a block.

In all test cases, we observed improved trunk detection results when operating with both the lidar and image data. With information obtained from pixelwise image classification, we can be more certain about the presence of a trunk and discard lidar candidates corresponding to overhanging foliage and/or tall grass. This improvement was more pronounced when operating on data collected during the preharvest season, with performance increasing from 48.1% to 93.5%. During preharvest, lidar-based detection returned significantly larger false-positives counts. Figure 13 illustrates the advantage of the multimodal approach with point cloud data of apple trees with medium foliage density. On the other hand, data scanned during the flowering season (with nearly bare trees) produced near perfect trunk detection (99% accuracy) using either method. In this case, the remaining 1–2% error was due to support poles being classified as tree trunks.

The trunk detection pipeline performed better on data from the Pink Lady apples (95.8% accuracy) than from the Granny Smith (86.8–95.3% accuracy). This performance difference was more significant within the lidar analysis, varying from 89.7% to 48.1% accuracy. The Pink Lady block had much less tall grass, because the farmer (according to their independent management practice) had placed a herbicide strip prior to the trial to clear the grass and weeds from around the base of the trees. As a result, there was less occlusion around the Pink Lady tree trunks in comparison to the Granny Smith trunks.

With data captured at a slower vehicle speed, the trunk detection pipeline performed well, with 97.5–99.3% accuracy when using both lidar and image data (Table II). The same was not observed if using lidar data alone (80.1% accuracy on the V-trellis structure), where we observed an increased number of false-positive returns. The denser point cloud data caused a greater number of lines to be fit to areas with overhanging foliage and/or tall grass. Finally, there was an insignificant difference in trunk detection accuracy with data captured over the two trellis structures.

4.1. Learned Observation Models

In Section 3.2.3, we introduced observation models [Eq. (4)] for both sensor modalities, which describe the likelihood of an observation for a given state. Although previously hand-crafted (Figure 12), we can now capture the true distributions using the ground truth position of the tree trunks. The ground truth data can be distributed along the slices, allowing us to infer the true state sequence along each row. The observations from lidar data (line length) and image data (number of pixels classified as trunks) for each slice, along with its true state, can be used to generate a histogram of state distribution over the observation range. From this we can evaluate the likelihood of a state given a particular observation, resulting in the learned observation models shown in Figure 15.

The relationship between the states and the observations matches our hand-crafted observation models [Figures 12(a) and 12(b)]. For image observations, there is a crossover in the likelihood of the trunk and gap states as the number of trunk-classified pixels increases. A similar crossover was applied within the hand-crafted model. The location of this crossover can be influenced by either the severity of foliage occlusion or the accuracy of pixelwise classification. For example, with perfect image classification and no trunk occlusions, this crossover would be pushed to the far left as any number of pixels classified as a trunk would infer the presence of a trunk state. Interestingly, a similar crossover is not present in the learned lidar model. This suggests that we are always more likely to be observing a gap rather than a tree, regardless of the lidar observation. If operating locally and in isolation, such an observational model would never detect a trunk. However, both the hand-crafted and learned observations express the important relationship that we are *more likely* to observe a trunk with longer line observations than with shorter observations (and the opposite for gaps). When combined with the constraint imposed by the duration model, the likelihood of a trunk state increases when the longer line segments are observed at approximately the expected interval.

By separating the learned models between the seasons, we can visualize the differences in how the environment is represented by the data. A longer line fit from the Hough transform is more likely to be a trunk when the data are collected from the flowering season [Figure 15(a)] than from the preharvest season [Figure 15(c)]. For the image observation

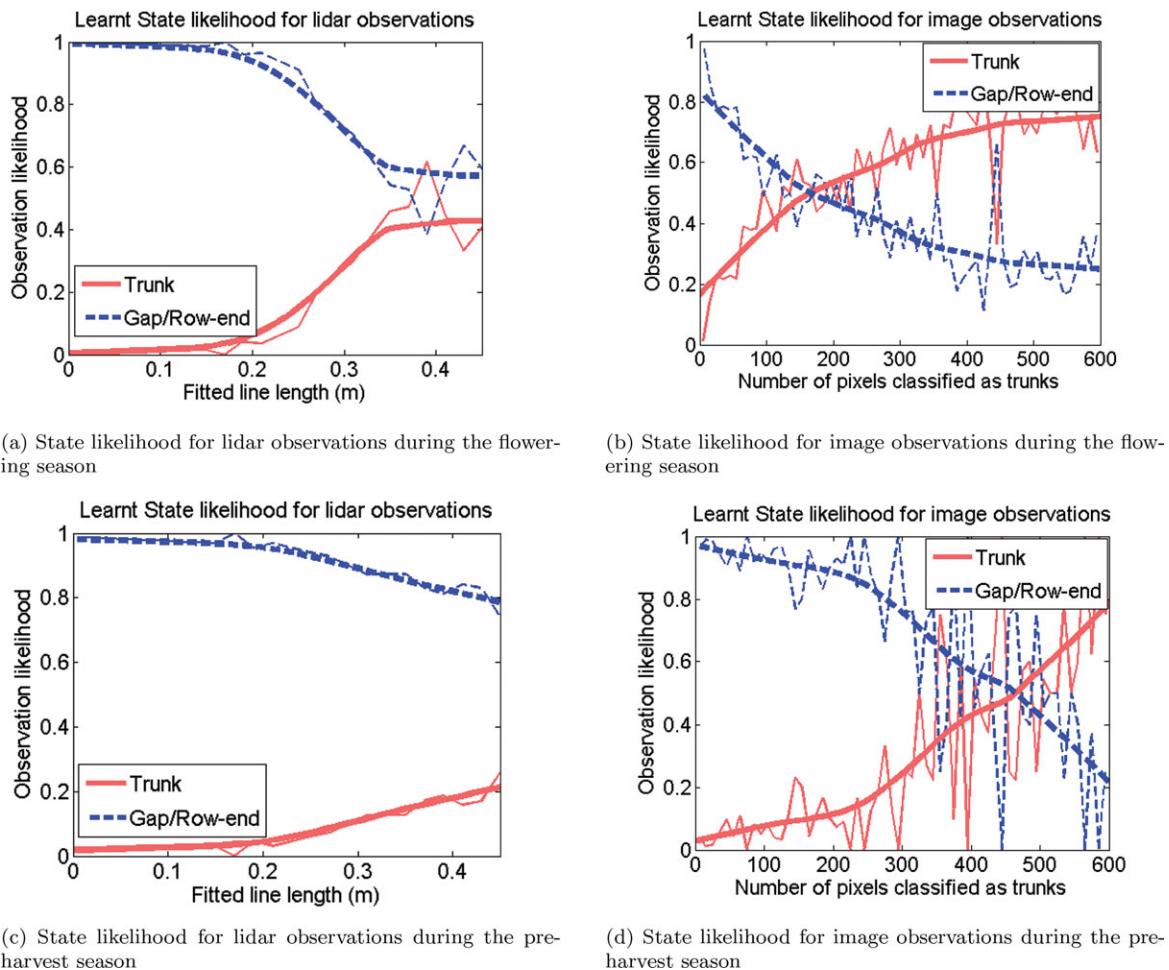


Figure 15. Empirically evaluated observation likelihoods for the image and lidar observation over the flowering and preharvest season. Each dataset is overlaid with a smooth interpolation using a moving average function.

model, the delayed crossover between the state likelihoods during the preharvest season suggests an increased number of foliage occlusions in the image data. The learned models from each dataset were fed back into the pipeline to reevaluate the detection accuracy. The results are summarized in Table III.

The use of the learned models resulted in minor improvements in detection accuracy in all cases (other than dataset 4, which was the same). These results may be considered optimal, though it should be noted that they were obtained from hand-labeled ground truth data (labeled trunk locations), which were generated for the performance evaluation in this paper and would not normally be available. The purpose of these results is to show that the intuitive, manual approach to constructing observation models yielded close to optimal results, so we can consider the process and the resulting models to be valid. Future work could examine the possibility of hand-labeling a small section of the

Table III. Trunk detection accuracies using hand-crafted vs learned observation models.

Dataset	Observation Model	
	Hand-crafted	Learned
1. Preharvest	95.8	95.9
2. Preharvest	86.8	89.5
3. Preharvest	93.5	95.8
4. Flowering	99.0	99.0
5. Flowering	99.0	99.1

orchard to learn the models, which could then be propagated to the rest of the data, but it is not known how much labeled data would be required to outperform the hand-designed models.

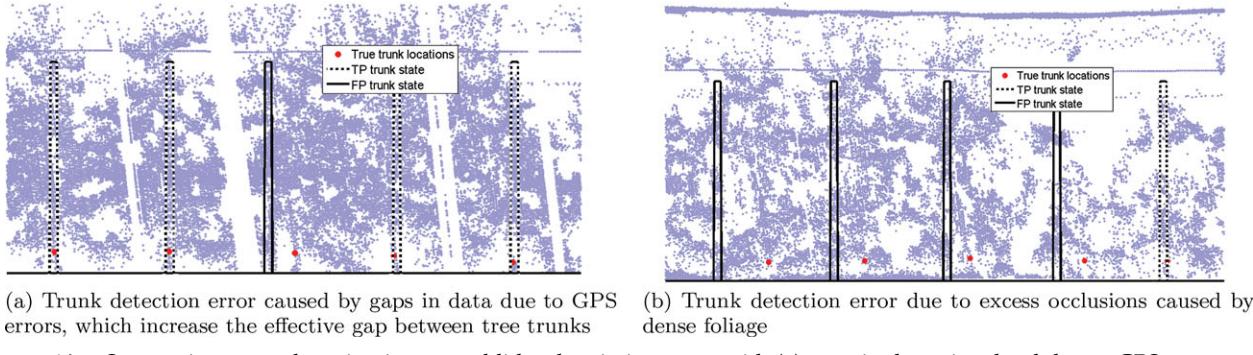


Figure 16. State estimate results using image and lidar data in instances with (a) gaps in the point cloud due to GPS errors, and (b) large amounts of low-hanging foliage.

4.2. Sources of Error

Most of the trunk detection errors were due to two primary causes. First, occlusions of the GPS satellite signal caused jumps in the vehicle position estimates, affecting the georeferenced point cloud. Such errors were more evident in areas with heavier foliage, i.e., during the preharvest season. This is shown in Figure 16(a), where the gap in the point cloud data increases the effective gap state duration between the neighboring tree trunks. In instances with smaller GPS errors, the global nature of the HSMM algorithm provided robustness against jumps in data by optimizing the state estimates over the whole row. However, in order to eliminate such problems altogether, vehicle odometry could instead be used for point cloud reconstruction along the row.

Secondly, even with the inclusion of image data, we observed misclassification in areas of the heaviest foliage/grass. Examples of such areas are shown in Figures 16(b) and 10(b). In such instances, it is difficult to tell the apple tree trunks apart even with the human eye. Lidar observations returned trunk candidates with very low precision (a linear fit made almost everywhere), and image classification was not able to detect the underlying tree trunks. Another source for false-positive detection was the vertical support structures at the farm. Although these instances of misclassification were rare, they occurred primarily within the I-trellis structures as the trunks are closer together.

5. DISCUSSION

The success of the pipeline for trunk detection over the two trellis structures, apple varieties, vehicle traversal speeds, and at different times of the year demonstrates the ability of the pipeline to generalize to different conditions at the apple orchard. The HSMM algorithm at the core of this pipeline has also proven to be adaptable to different orchards. Operating on a citrus orchard where the trees are planted in squat formation, Wellington and Campoy (2012) reported a 99% tree detection accuracy using this HSMM framework. Their performance measure is based on using detection for

tree counting and comparing against the farm's true tree count (total number of trees). However, any overcounts and undercounts along the row are not treated separately while tallying, meaning that such errors tend to cancel out on average. A similar segmentation technique was also utilized in Jagbrant et al. (2014), who instead evaluated the tree detection rate (as done in this paper), reporting an accuracy of 99% on an almond orchard. By comparison, the intertwined and cluttered nature of trellis structured orchards poses significant detection challenges that have been tackled by the pipeline presented in this paper.

5.1. Information Management System

To incorporate the detected trunks within a complete information management system, each unique tree must be *uniquely identified* or located. Once constructed, such a tree inventory can be used to store and manage data relating to each tree, and the database can be updated on subsequent scans. Trees can be uniquely located based on their metric position if derived from GPS/odometry, or they can be labeled in a topological manner (e.g., row 15, tree 1, 2, ..., 35, 36...) by sequentially counting the detections along each row.

If using GPS, each trunk can be uniquely identified by matching its global position. However, GPS is not sufficiently accurate to isolate an individual tree while operating in between the tall tree structures. A tree map constructed using the GPS/INS navigation system is shown in Figure 3(f), where actually straight rows are distorted due to erroneous localization. A more suitable solution would be to use odometry to localize data within a row and then either have physical markers or GPS localization to mark specific rows from the headland.⁵ This is, however, not within the scope of this paper.

⁵GPS is typically more accurate at the headlands of an orchard due to increased visibility of the sky.

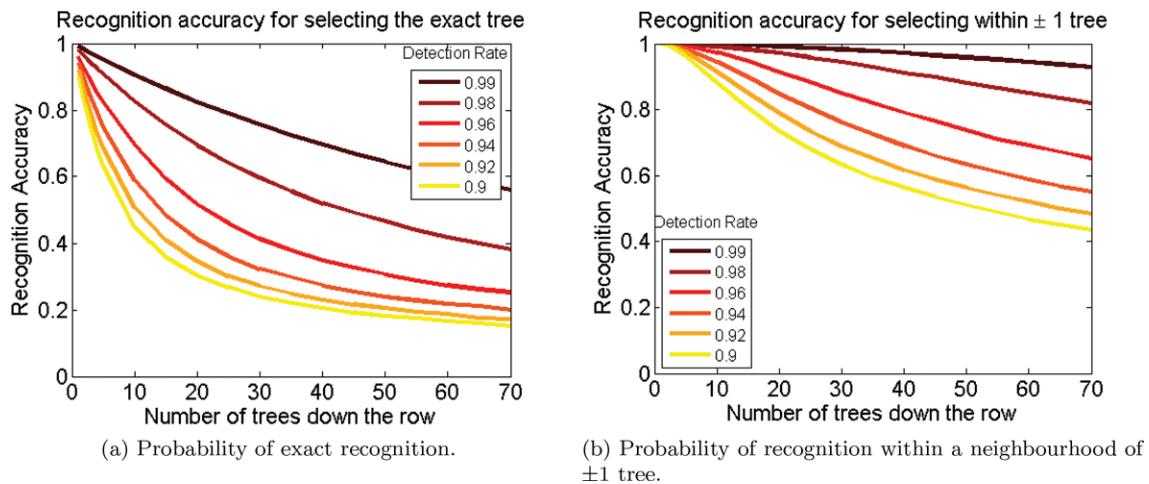


Figure 17. Recognition accuracy achieved by sequential counting for different detection accuracies.

Purely topological trunk localization (e.g., row 15, tree 23) done with sequential counting requires near perfect detection rates because the counting error accumulates over the length of the row. A Monte Carlo simulation can help model the requirements of a detection and counting-based approach for a topological tree inventory. By progressively simulating tree counting down the length of a row while considering the stochastic detection for each tree, we can infer the accuracy of uniquely and correctly identifying each individual tree as a function of its position (sequential depth) within each row. Figure 17(a) illustrates the probability of locating a particular tree via sequential counting for different detection rates. Figure 17(b) shows the same analysis with the recognition requirements relaxed to a neighborhood of ± 1 tree. We can see that due to the accumulative nature of counting errors, the recognition accuracy degrades rapidly along the length of the row. For example, the probability of finding the 30th tree within ± 1 trees with a detection accuracy of 96% is only 85%, whereas the probability of identifying the exact tree is only 41%. This suggests that detection can be reliable for identifying trees in the field and within the data, but a secondary system is needed to recognize specific trees. An intermediate solution to reduce infrastructure cost when considering individual tree tags could be to use RFID or QR tags sparsely. This will constrain the upper bound of recognition errors by periodically zeroing the sequence error. The spacing of the tags would be a function of their cost, the required recognition accuracy, and the accuracy of the trunk detection method. An infrastructure-free solution is given by Jagbrant et al. (2014), where the appearances of the trees are used to uniquely identify them. The detection method presented here is required as a module within any of these solutions.

5.2. Lessons Learned

The main lessons learned through practical application of this pipeline are outlined as follows:

- The results suggest that in order to acquire the most accurate tree detections spanning over an orchard block, the trunk segmentation pipeline should be applied on data collected during the flowering season, or carefully timed with the use of herbicide strips to reduce undesired foliage. Furthermore, if operating strictly within the flowering season, a lidar-only detection pipeline can provide near perfect results (98% detection accuracy).
- Data captured at slower speeds produce a denser point cloud, which provides a denser representation of the tree trunks, but it also results in more returns from foliage clutter. These are often skipped by the sensor at higher speeds. Ideally, the vehicle speed should be a function of the trunk width and the frame rate of the lidar sensor in order to go as fast as possible without skipping the trunks.
- The trunk detection performance is not sensitive to the exact parametrization of the models within the HSMM framework. A two-step inference process can be used to update the duration model, where we learn the duration function from iteration and then feed it back into the framework. Regardless, intuitive and easily interpretable hand-crafted observation models can be designed to produce near-optimal trunk detection. Learned models can be developed through hand-labeling to check the validity of the hand-crafted models and to visualize differences in how the environment is represented by the data.

- Detection accuracy could be improved if trunk occlusions from the overhanging foliage could be further minimized by optimizing the sensor configuration. For example, optimal frame-rates and vehicle speeds could be chosen, and the lidar and image sensors could be mounted closer to the ground for a more suitable trunk viewing angle.
- The camera was mounted relatively high on the vehicle and pointed down to the ground and trunks. Due to this, light artifacts such as lens-flares within the image data were minimized. Any image classification errors that did occur could be captured within the observation model and then filtered out through the HSMM. This allowed us to work under variable natural illumination conditions, increasing the hours during the day when we could operate.

6. CONCLUSION

We have described a pipeline that uses a probabilistic approach to detect trees at an orchard configured with dense trellis structures. The pipeline processes raw lidar and image data in order to detect individual trunks, representative of trees at the orchard. This includes automatic row extraction from lidar point cloud data, building a trunk observation model using the lidar data, updating the model through image analysis, and using a hidden semi-Markov model for trunk detection, which leverages from the regular structure of the orchard.

The pipeline performance was tested on two different trellis structures (I-trellis and V-trellis) at different times of the year (flowering and preharvest), with different apple varieties, and with different vehicle traversal speeds. It was adaptable to all of these variations, producing good performance on data obtained during the preharvest season (95.8% accuracy) and a near perfect tree detection (99% accuracy) with data obtained during the flowering season. Furthermore, hand-tuned parameters, which are intuitive and easy to design according to what the engineer knows about the problem domain, provide near-optimal detection accuracy.

This work is directed toward the development of an information-gathering and -processing system, where trunk detection can be combined with localization or recognition to produce tree inventories. This will allow for efficient management of crop data down to the individual tree. Future directions will therefore need to focus on developing reliable localization systems, which could provide for either a metric or topological representation of the trees. Odometry can be used to localize detected trunks within a row, reducing the reliance on a GPS system. Additionally, lidar- or image-based descriptors for each tree can be used to perform sequence-based orchard-wide tree recognition and localization, as was previously done at an almond orchard in Jagbrant et al. (2014).

ACKNOWLEDGMENTS

This work is supported by the Australian Centre for Field Robotics at The University of Sydney, and Horticulture Australia Limited through project AH11009 Autonomous Perception System for Horticulture Tree Crops. Thanks to Gustav Jagbrant for his contribution in the implementation of the HSMM algorithm, and to Kevin Sanders for his support during the field trials at his apple orchard. Additional information and videos are available at <http://sydney.edu.au/acfr/agriculture>.

REFERENCES

- Bargoti, S., Underwood, J. P., Nieto, J. I., & Sukkarieh, S. (2014). A pipeline for trunk localisation using lidar in trellis structured orchards. *Field and Service Robotics* (pp. 455–468). Springer.
- Bhattacharya, P., Liu, H., Rosenfeld, A., & Thompson, S. (2000). Hough-transform detection of lines in 3-D space. *Pattern Recognition Letters*, 21(9), 843–849.
- Bilmes, J. A., et al. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute*, 4(510), 126.
- Christensen, E. (2012). From orchard to market: Come along on an apple harvest. Retrieved January 12, 2006, from <http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software>.
- Dey, D., Mumtaz, L., & Sukthankar, R. (2012). Classification of plant structures from uncalibrated image sequences. In *Proceedings of the Workshop on Applications of Computer Vision (WACV)* (pp. 329–336). IEEE.
- Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., & Frenkel, A. (2011). On the segmentation of 3D LIDAR point clouds. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (pp. 2798–2805). IEEE.
- Farabet, C., Couprie, C., Najman, L., & Lecun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)* (pp. 575–582).
- Hamner, B., Singh, S., & Bergerman, M. (2010). Improving orchard efficiency with autonomous utility vehicles. In *American Society of Agricultural and Biological Engineering Annual Convention Proceedings Abstract and Paper* (Vol. 28).
- Hung, C., Nieto, J., Taylor, Z., Underwood, J., & Sukkarieh, S. (2013). Orchard fruit segmentation using multi-spectral feature learning. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)* (pp. 5314–5320). IEEE.
- Illingworth, J., & Kittler, J. (1988). A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1), 87–116.

- Jagbrant, G., Underwood, J. P., Nieto, J., & Sukkarieh, S. (2014). Lidar based tree and platform localisation in almond orchards. *Field and Service Robotics* (pp. 469–483). Springer.
- Jimenez, A., Ceres, R., Pons, J., et al. (2000). A survey of computer vision methods for locating fruit on trees. *Transactions of the American Society of Agricultural Engineers*, 43(6), 1911–1920.
- Lalonde, J.-F., Vandapel, N., Huber, D., & Hebert, M. (2006). Natural terrain classification using three-dimensional Ladar data for ground robot mobility. *Journal of Field Robotics*, 23(10), 839–861.
- Moorehead, S. J., Wellington, C. K., Gilmore, B. J., & Vallespi, C. (2012). Automating orchards: A system of autonomous tractors for orchard maintenance. In *International Conference on Intelligent Robots and Systems (IROS)*, Workshop on Agricultural Robotics.
- Munoz, D., Bagnell, J. A., & Hebert, M. (2010). Stacked hierarchical labeling. In *Proceedings of the 11th European Conference on Computer Vision: Part VI, ECCV'10* (pp. 57–70). Berlin, Heidelberg: Springer-Verlag.
- Nielsen, M., Slaughter, D. C., Gliever, C., & Upadhyaya, S. (2012). Orchard and tree mapping and description using stereo vision and lidar. In *International Conference of Agricultural Engineering*.
- Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., & Singh, S. (2014). Automated visual yield estimation in vineyards. *Journal of Field Robotics*, 31(5), 837–860.
- Pauling, F., Bosse, M., & Zlot, R. (2009). Automatic segmentation of 3d laser point clouds by ellipsoidal region growing. In *Australasian Conference on Robotics and Automation (ACRA)*.
- Subramanian, V., Burks, T. F., & Arroyo, A. A. (2006). Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. *Computers and Electronics in Agriculture*, 53(2), 130–143.
- Tumbo, S. D., Salyani, M., Whitney, J. D., Wheaton, T. A., & Miller, W. M. (2002). Investigation of laser and ultrasonic ranging sensors for measurements of citrus canopy volume. *Applied Engineering in Agriculture*, 18(3), 367–372.
- Underwood, J., Scheding, S., & Ramos, F. (2007). Real-time map building with uncertainty using colour camera and scanning laser. In *Australasian Conference on Robotics and Automation*.
- Vandapel, N., Huber, D. F., Kapuria, A., & Hebert, M. (2004). Natural terrain classification using 3-d ladar data. In *Proceedings of the International Conference on Robotics and Automation (ICRA)* (Vol. 5, pp. 5117–5122). IEEE.
- Wei, J., & Salyarii, M. (2005). Development of a laser scanner for measuring tree canopy characteristics: Phase 2 foliage density measurement. *Transactions of the American Society of Agricultural and Biological Engineers*, 48(4), 1595–1601.
- Wellington, C., & Campoy, J. (2012). Orchard tree modeling for advanced sprayer control and automatic tree inventory. In *International Conference on Intelligent Robots and Systems (IROS)*, Workshop on Agricultural Robotics.
- Yu, S.-Z. (2010). Hidden semi-Markov models. *Artificial Intelligence*, 174(2), 215–243.
- Yu, X., Hyyppä, J., Kaartinen, H., & Maltamo, M. (2004). Automatic detection of harvested trees and determination of forest growth using airborne laser scanning. *Remote Sensing of Environment*, 90(4), 451–462.
- Zhang, J., Chambers, A., Maeta, S., Bergerman, M., & Singh, S. (2013). 3D perception for accurate row following: Methodology and results. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)* (pp. 5306–5313).