A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture

Luis Emmi, Emile Le Flecher, Viviane Cadenat, Michel Devy

A hybrid representation of the environment to improve autonomous navigation of mobile robots in agriculture

L. Emmi[1] · E. Le Flécher[1] · V. Cadenat[1,2] · M. Devy[1]

Abstract
This paper considers the problem of autonomous navigation in agricultural fields. It pro-poses a localization and mapping framework based on semantic place classification and key location estimation, which together build a hybrid topological map. This map bene-fits from generic partitioning of the field, which contains a finite set of well-differentiated workspaces and, through a semantic analysis, it is possible to estimate in a probabilistic way the position (state) of a mobile system in the field. Moreover, this map integrates both metric (key locations) and semantic features (working areas). One of its advantages is that a full and precise map prior to navigation is not necessary. The identification of the key locations and working areas is carried out by a perception system based on 2D LIDAR and RGB cameras. Fusing these data with odometry allows the robot to be located in the topological map. The approach is assessed through off-line data recorded in real condi-tions in diverse fields during different seasons. It exploits a real-time object detector based on a convolutional neural network called you only look once, version 3, which has been trained to classify a considerable number of crops, including market-garden crops such as broccoli and cabbage, and to identify grapevine trunks. The results show the interest in the approach, which allows (i) obtaining a simple and easy-to-update map, (ii) avoiding the use of artificial landmarks, and thus (iii) improving the autonomy of agricultural robots.

Keywords Hybrid topological map · Crop classification · Semantic identification · Autonomous navigation · Agricultural robotics

✉ L. Emmi
luis.emmi@laas.fr

✉ V. Cadenat
cadenat@laas.fr

[1] CNRS-LAAS, Université Toulouse, 7 Avenue du Colonel Roche, F-31077 Toulouse CEDEX, France

[2] Université Toulouse, UPS, LAAS, F-31400 Toulouse, France

Introduction

In the last 20 years, a large number of robots have been developed to help in daily tasks (Bergerman et al. 2016; Royakkers and van Est 2015), including the management and care of crops. The design of research prototypes and commercial robots for agriculture has been recently growing (Ampatzidis et al. 2017; Gonzalez-de-Santos et al. 2017). Their autonomy has also been improved, reducing the need for human intervention. Many of these developments focus on solving the problem of navigation within the cultivated fields of a farm. There are two main approaches for robot location (Bechar and Vigneault 2016; Kanagasingham et al. 2020): absolute location, mostly used in global navigation satellite system (GNSS) navigation (Kayacan et al. 2015; Keskin et al. 2017), and relative location, mostly exploited in sensor-based navigation (Malavazi et al. 2018). The first kind of strategy relies on global metric data commonly provided by real-time kinematic GNSS (RTK-GNSS). These data are used to build a metric map. This map is required prior to navigation and is used to geographically locate robots in the field. Although this approach is widely used in agricultural applications, some drawbacks can be identified. First, the high cost in terms of equipment (receivers and antennas) and fees (subscriptions to RTK networks) represents, in some cases, a considerable cost compared to the total cost of the robots and the application. Second, GNSS data can only be used to build metric maps that are difficult to keep up to date because agricultural fields are dynamic environments that evolve according to seasons. Last, its vulnerability in certain areas (GNSS-denied zones such as orchards, for example) may lead to location inaccuracies, which in turn may induce task failures.

Sensor-based navigation allows these issues to be addressed. It relies on onboard proximity sensors, such as cameras or LIDARs. It appears to be efficient when the GNSS signal is poor or unavailable. Indeed, it has been widely used (i) to follow rows (Blok et al. 2019; Comba et al. 2010; García-Santillán et al. 2018), (ii) to safely perform U-turns (Durand-Petiteville et al. 2017), and even (iii) to avoid obstacles in different contexts (Chen and Tsai 2000; Cherubini et al. 2014). Although this topic has been addressed in recent years (Vougioukas 2019), its major drawback is the impossibility to navigate large distances without a map due to the limited range of the onboard sensors that do not allow the final goal to be observed at any time. Therefore, information about the structure and distribution of the working environment is required, and it is generally provided by the user in a prior operation, making most of the solutions specific to the field under consideration. For example, some solutions consist of adding artificial landmarks to identify stopping or transition conditions (Li et al. 2010) to choose the best actions to perform depending on the context (e.g., when the end of a row is detected, a U-turn should be executed).

An alternative, commonly used in mobile robotics for solving the lack of a full map in sensor-based approaches, is the application of topological maps. A topological map is a discrete description of the working area and is applied to estimate the state/location of a robot in partitioned and semi-structured environments (Kostavelis and Gasteratos 2015; Thrun 1998). This representation is built as a graph containing nodes and arcs. In robotic navigation, the nodes represent an abstraction of physical areas of the environment, while the arcs define the conditions to change between nodes. This type of diagram has been simplified so that only vital information remains, and unnecessary details are removed, i.e., they lack scale and distance, and the nodes illustrate an array of user-defined parameters (Lowry et al. 2015). Nevertheless, this type of approach has not been intensively applied in the agricultural domain and can provide solutions to present challenges in the fields of localization and autonomous navigation.

Although some proposals have approached this situation in recent years (Hague et al. 2000; Shamshiri et al. 2018), there is still a need for general navigation strategies that are less dependent on (i) human intervention, (ii) absolute location, and (iii) a controlled environment, allowing long-range navigation to be achieved. This problem is the foundation of this paper and is realized in the context of a collaborative project that aims to improve the autonomy of the Oz platform, a commercial weeding robot developed by Naïo Technologies (Naïo 2020). Indeed, the onboard navigation strategy running on this robot requires the use of artificial landmarks (red sticks) for row boundary detection. In addition, it still cannot recognize and identify the crops present in the field. Thus, this paper offers the first step in fulfilling these objectives. It presents a location and mapping framework that takes advantage of the particular structure of the field, and by extracting natural landmarks, it can locate the robot in a topological map. The presented framework consists of a perception system (LIDAR and RGB camera) that acquires and labels the information, which is then analyzed by a semantic classifier to extract natural features. The perception system and the semantic classifier are accompanied by a map update process. The proposed framework is validated through off-line data recorded in real conditions in diverse fields during different seasons with two robotic platforms: Oz and Aircobot (Futterlieb et al. 2014).

Background and related work

Currently, there are numerous companies that offer diverse mapping services, from delimiting the farm zones to obtaining in detail several vegetation and soil indices. The mapping task is primarily performed by the use of unmanned aerial vehicles (UAVs), satellite images or manually (Adão et al. 2017; Xue and Su 2017). Most solutions use dedicated software to analyze the data and mainly rely on RTK-GNSS. Moreover, most of the maps that are created in these solutions use the contour of the field given by an external source or acquired with their systems and generate the path that the tractor driver should follow for optimal operation. These are called full coverage maps, which do not consider the crop rows, the planted area or the headland. Most of the maps that are available via this kind of service provide some information for supporting autonomous farm navigation by mobile robots, such as seeder-extracted maps. However, the acquired data need to be manually processed to be adapted for (i) a specific robotic system, (ii) a specific task, and (iii) a specific environment.

One alternative to absolute mapping is the fusion of proximity sensors (i.e., LIDAR) with odometry and/or camera systems, which has been a broad subject of research (Penizzotto et al. 2015; Thanpattranon et al. 2016). Most of the studies have sought to solve the problem of row following (Sharifi and Chen 2015; Tu et al. 2014), and only some of them addressed detecting the beginning and end of the crop row. Generally, artificial landmarks are installed for the detection of these limits (Li et al. 2010; Zhang et al. 2014). Although this is a suitable solution for perennial crops, manual work is required for their installation and removal in non-perennial crops. This limits the autonomous capabilities of robotic systems. An alternative is using the information already provided by the natural environment through the semantic extraction of key elements and adequate modeling of the working environment (Blanke et al. 2012; Bochtis et al. 2010). Related to this subject, a specific work where semantic classification of a field was proposed has aroused interest (Weiss and Biber 2010). In this study, a particle filter was used to determine the probability

distribution of the robot state in the different locations of the field. It should be noted that in this approach, (i) there were no limits on the meaning of the open field, (ii) the row-start and row-end areas were not specific points, and (iii) the difference between row-start and row-end areas was not clear, which hinders its use for accurately defining the transition between rows and an open field.

The field model

In general, a field contains one type of crop, planted in parallel and at a relatively constant distance between plants. Moreover, a field can be partitioned into four places: headland, lane, alley and gate (see Fig. 1a). The alley and lane represent the cultivated area, with particularity for the alley, which also corresponds to the lateral boundaries of the field. Both lanes and alleys can include more than one inter-row space. Moreover, their width corresponds to the width of the tool/implement. The headland is the area where the robot switches between lanes, alleys and the gate. Finally, the gate allows the robot to enter/quit the field through the headland. For this representation, the gate is a way to arrive from the farm at a specific part of the headland, and its location depends on the peculiarity of every field and farm. The capability to divide the field in a limited number of places allows the problem to be described by using a topological map (see Fig. 1). One of the advantages of using this type of map is the ability to reduce the cumulative error due to relative sensors by validating the abstraction from a continuous physical world to a discrete topological network description (Kuipers and Byun 1991). For these reasons, a topological map is less sensitive to variations in the field.

Consequently, the working space partition represents the possible robot states (see Fig. 1b), i.e., a semantic representation of the location of the robot in the field. A fifth state called 'unknown' has been included to handle the uncertainties when sensory data are not conclusive. A state transition toward this particular state (dotted gray arrows, see Fig. 1b) indicates that the system is lost, so strategies must be applied to be located again on the map. It should be noted that state transitions are assumed to occur when the robot navigates forward, given that backward motion occurs on exceptional occasions. Successful navigation is not dependent on geometric accuracy, given that the control and topology levels do not depend on the geometric description. However, the sole use

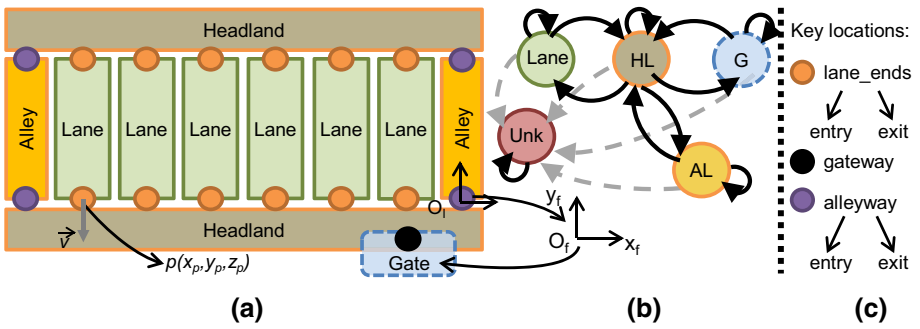

Fig. 1 The proposed field model. **a** The semantic field place partitioning. **b** Topological representation (topological map) of the workspaces and allowed transitions (lane; *HL* headland; *G* gate; *AL* alley; *Unk* unknown). **c** Key location categories

of abstract representation of the working environment does not allow a required precision for this type of system to be attained. Therefore, the semantic representation only allows the system to instantaneously distinguish in which type of place (state) the robot is present, but it cannot estimate the exact position where the transitions are located with respect to the current robot position. This kind of map may not be sufficiently accurate for efficiently controlling the tools/implements, given that the identification of the beginning and end of the rows ends up being a fuzzy area. In this context, the inclusion of geometric information allows route planning or the resolution of topological ambiguities to be achieved. This is the particular case of the transitions between nodes. For this reason, a metric map is included by estimating the accurate co-ordinates of the transitions, called key locations (see Fig. 1c), and can be categorized as follows: *lane_ends*, *alleyways* and *gateway*. Moreover, depending on whether the key location has been estimated from the headland or from the other places, they can be classified as type entry or type exit, respectively.

To build a metric map, the first step is to define the co-ordinate system that represents it. Since this representation of the field does not require an absolute co-ordinate system, the field co-ordinate frame is defined as $F_f = \left(O_f, \vec{x}_f, \vec{y}_f, \vec{z}_f\right)$ (see Fig. 1a), where origin O_f is located at the gate position, or failing that, in the initial position of the system. As introduced above, the elements that constitute the metric map are the co-ordinates that represent the transitions between states, called key locations. Since the metric map is built based on the movement of the system in the field and the range of the perception system is not sufficient for observing the whole field at once, the key locations must be constantly generated, representing possible transitions between states. When a transition is validated, the most likely key location becomes part of the metric map. An auxiliary reference frame $F_l = \left(O_l, \vec{x}_l, \vec{y}_l, \vec{z}_l\right)$, related to F_f, is used to keep the perception system information referenced close to the last key location identified in a state transition. This temporary reference frame (F_l) is relocated whenever a state transition is detected, which generates a flush of the stored information of the perception system. This allows the drift caused by odometry in long-range navigation to be minimized.

Therefore, the key locations are represented by poses $Kloc_t^s\left(p, \vec{v}\right)$, where $p\left(x_p, y_p, z_p\right) \in \mathbb{R}^3$ represents the position of the key location with respect to the field co-ordinate frame F_f (see Fig. 1a), and $\vec{v} \in \mathbb{R}^3$ is a geometric vector with a unitary magnitude that defines the orientation of p in F_f. Moreover, t is a notation for classifying the type of the key location, which can be an entry (*en*) or an exit (*ex*). The notation s represents the side where the key location has been identified with respect to the instantaneous position of the robot. It can then have the value left (l), right (r) or middle (m). Being able to classify from which side the key location has been estimated gives additional information for defining its category and validity (see Fig. 1c). If right and left are identified, both key locations can be fused to create a middle key location. For convenience, the position p of the key location is located in the center of the lane or alley (see Fig. 1a) and only in the transition between workplaces. This is done regardless of whether the point was identified to the left or right of the instantaneous position of the robot. The point p is moved to the center of the row (in the case of the lane) or moved perpendicularly to the opposite side (in the case of the alley). Moreover, $\vec{v}$ represents the orientation of the lane or alley that corresponds to said key location with respect to F_f. For this problem, the gateway and, consequently, the gate are not considered for both the estimation of key locations and for the location of the robot in the field, given that the gate representation depends on the peculiarity of each field and the farm.

The instantaneous position of the robot based on the odometry is defined as $Rob\left(r, \vec{h}\right)$, where $r(x_r, y_r, z_r) \in \mathbb{R}^3$ represents the position of the center of the robot with respect to the field co-ordinate frame F_f (see Fig. 1a), and $\vec{h}$ is a geometric vector with unitary magnitude that defines the orientation of r in F_f, i.e., the heading of the robot. To simplify the problem, $z_p = z_r$ is considered, assuming that the effect of the slope of the field on the key location estimation can be ignored, considering that the field of view of the perception system onboard the robot is narrow (only a few meters).

The presented framework includes general modeling of the field expressed throughout a hybrid topological map. A semantic classifier is developed to compare the onboard sensor information with the field model and thus estimate the place where the robotic system is currently located. Moreover, by estimating the key locations in a metric reference frame, it is possible to accurately define where the transitions occur. Both elements build a hybrid topological map. Several advantages can be highlighted for this type of map: (i) it captures the environment connectivity while limiting the use of metric information, (ii) the map is 'lighter' and easier to update, (iii) absolute location sensors and previous mapping are no longer necessary, and (iv) artificial landmarks are no longer required.

Algorithm for semantic location and mapping

Perception system: data acquisition, labeling and clustering

Figure 2 presents the developed algorithm. The algorithm looks for key locations *Kloc* by analyzing the distribution of the crops on the field, encompassed in clusters of laser points.

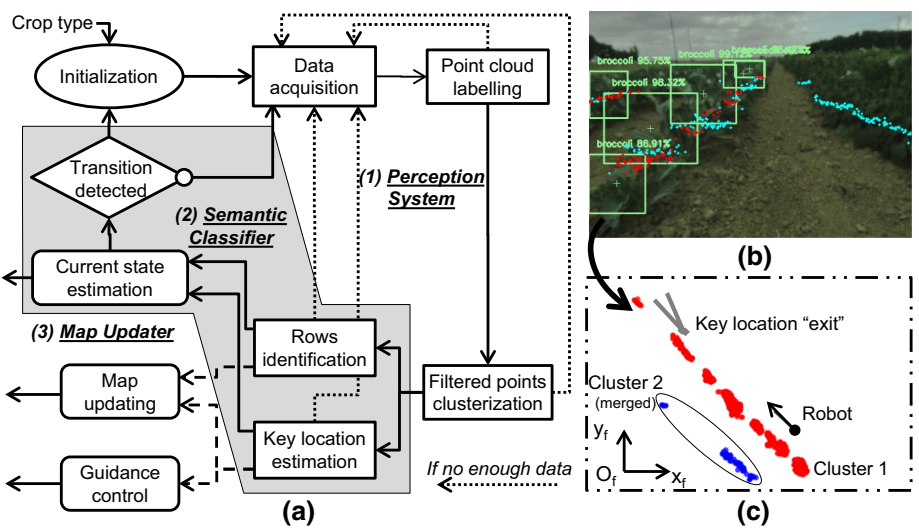

Fig. 2 Algorithm for semantic location and mapping. **a** Logical scheme for state estimation and key location estimation. **b** Image of the left camera of the Oz platform in a broccoli field, showing the laser points labeled (red points) based on crop detection and all laser points (cyan points). **c** Projection of the laser points in the Cartesian plane F_f', showing the clustered points labeled in **b**, the key locations exit estimated and the robot pose

Each of the laser points is referenced based on the field co-ordinate frame F_f. The type of crop is the sole required configuration input, and the laser points are labeled depending on their belonging crops (see Fig. 2b). The proposed crop classification algorithm requires a perception system made up of at least an RGB camera and a 2D laser scanner, which share the same field of view and are placed in front of the robot. The classification of the laser point requires three steps: (i) an image-based classification of the crops, (ii) a fusion between laser points and the image frame, and (iii) a filter that classifies the laser points belonging to crops detected in the image. The laser scanner should have enough clearance to be able to hit the crop, considering the type of crop and the growth stage, among other specific considerations. In conjunction with this perception system, the odometry of the robot is required to improve the robustness of the estimation of the key locations.

To identify the target crop from the rest of the environment (other fields, background, natural foliage, etc.), a great variety of techniques can be found in the literature. Crop detection by using RGB cameras has already been addressed over many years (Hamuda et al. 2016). These techniques are subject to variations in lighting conditions, and finding the appropriate threshold for each type of crop is a challenge that requires substantial effort to find and adjust. A less widespread approach is feature extraction within images (Shi et al. 2019). This makes it possible to identify and classify plants based on shape, stems, leaves, flowers, etc. Once the characteristics are extracted, machine learning algorithms are used to make the link between the observed characteristics and plant species (Wäldchen and Mäder 2018). These techniques have proven useful for plant phenotyping, but they require images where the crop is isolated from the background. Recently, it has been possible to find crop classification applications that use convolutional neural networks (CNNs) in which, unlike the methods presented previously, the features are not extracted during pretreatment (Lee et al. 2017; Potena et al. 2016). The advantage of this approach lies in its complexity, which makes it possible to manage a very large number of classes. The usage of this method is less common in phenotyping but more interesting in agriculture.

Among the alternatives of classifiers through CNN, two can be highlighted: object detection models and image segmentation models. The first can build a bounding box corresponding to each class in the image, while the second creates a pixel-wise mask for each object. The pixel-wise segmentation technique allows for a far more granular understanding of objects in an image with the disadvantage that annotating each class requires a greater effort than the bounding box technique. Since the main objective of this work is to carry out a simple but efficient crop classification and given the complexity of the objects to be trained (the crop with its intricate forms), the bounding box method is chosen.

Therefore, to perform crop identification, a comparative study was first carried out by analyzing recently developed object classifiers. Among the classifiers studied, two open-source CNNs stand out in terms of performance and are commonly used for object detection (Zhao et al. 2019): YOLO (Redmon and Farhadi 2018) and RetinaNet (Lin et al. 2017). Both architectures, including different configurations, were trained for classifying and detecting a set of market-garden crops. The results obtained (see "Results" section) show that the most appropriate classifier is YOLO version 3, mainly due to its real-time performance and its precision. Furthermore, YOLO version 3 does not require incorporating extra or expensive hardware, which can be advantageous when developing small agricultural robots.

Once the crops are detected in the images, only those belonging to the target crop are kept. Then, the laser points are transformed into an image reference frame, and the points that are within the positive detection region (i.e., inside the bounding boxes) are labeled (see Fig. 2b). Only these labeled points are used to describe the structure of the field. One

of the main advantages of this classification is that it properly identifies the alleys and the headlands, especially in the case of navigating close to different fields (diverse crops). Since the labeling process requires a certain computational load, several epochs of laser data are labeled from a single camera image. The labeled points are expressed as F_l based on the current robot pose *Rob*. Then, the points are clustered based on DBSCAN (density-based spatial clustering of application with noise) (Schubert et al. 2017) and stored in a point cloud, defined as *Pcl*. This point cloud is composed of laser points $Lp(l, c)$, where $l(x_l, y_l, z_l) \in \mathbb{R}^3$ represents the position of the laser point with respect to the field co-ordinate frame F_f, and c represents the number of clusters to which this point belongs. The purpose of carrying out this clustering methodology is to isolate each of the presented rows. Therefore, each cluster represents a single crop row, or part of a row, given that gaps can be found. Each cluster is analyzed by applying a linear regression and evaluating whether two clusters belong to the same row. If this occurs, both clusters are merged (see Fig. 2c). The clusters are stored until the system detects a change in the state. Therefore, every time new data are acquired, the stored clusters are updated. When a change in state is detected, the process is restarted, i.e., the clusters are discarded, and a new frame F_l is defined. Moreover, the laser points that are far behind the robot are also discarded to reduce the drift caused by the odometry. This procedure provides enough information to enrich the key location detection process, even if the robot location system is noisy.

Semantic classifier

The given point cloud *Pcl* and the current robot pose *Rob* are used to perform a semantic analysis to estimate the location of the robot in the topological map, denoted by *Est*. For this work, only the identification of the *headland*, *lane* and *alley* states are considered. Moreover, the *unknown* state is also considered and occurs when there is not enough information from the perception system to perform an identification of the working place. For this problem, the *gate* is discarded, given that its location and representation depend on the peculiarity of every field and farm. Thus, the instantaneous estimated robot location is given by

$$Est = \{(Pcl, Rob, n) | n \in N = \{headland, lane, alley, unknown\}\} \tag{1}$$

This analysis considers how the field is modeled and how the shape and orientation of the point cloud match that model. Therefore, the semantic classifier is carried out in three phases: (i) based on the classified laser points (*Pcl*), the key locations (*Kloc*) are estimated, and the information of the rows is extracted; (ii) with the row information, the location of the robot in the topological map is estimated; and (iii) considering the estimated state and the key locations, the state transition is evaluated.

Row identification and key location estimation

From the beginning of the navigation, the algorithm is constantly searching for both entry and exit types of key locations, including left and right ($Kloc^l_{en}, Kloc^r_{en}, Kloc^l_{ex}, Kloc^r_{ex}$), with respect to the current robot pose *Rob* (see Fig. 3). This search uses the relationship between the clustered point cloud *Pcl* with respect to the robot pose *Rob*, where the information of the lane width and row orientation is extracted. At the initialization, the lane width is defined as a priori knowledge and is updated based on the estimated key locations.

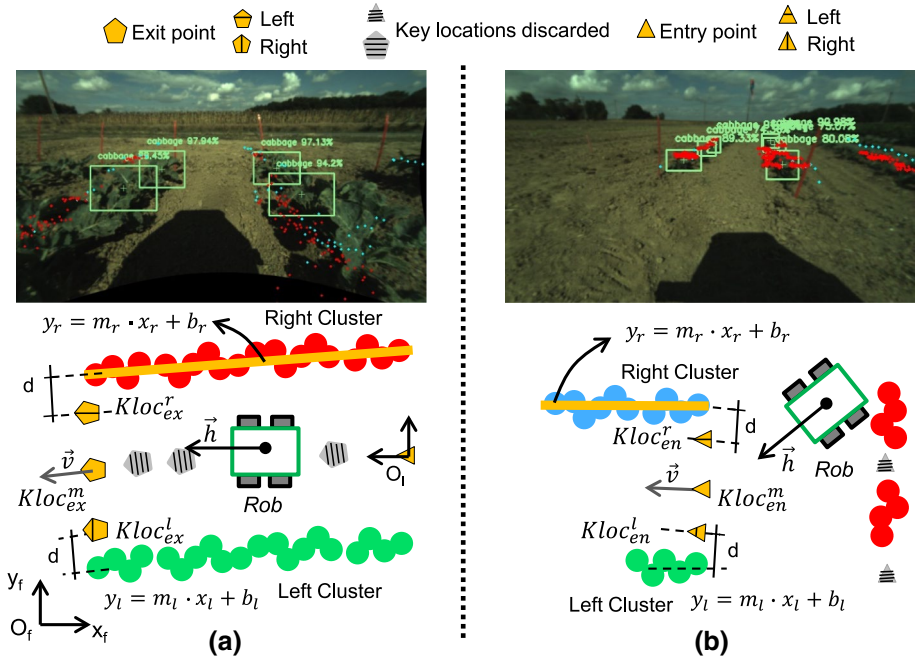

Fig. 3 Example of key location extraction for both entry and exit scenarios. In both cases, an example of an image extracted from the left camera of the Oz platform is presented showing the laser points labeled in a cabbage field. **a** Looking for the exit scenario: left and right key locations are calculated by projecting the furthest point of clusters left and right at *d* distance, respectively. If both key locations (left and right) are found, a middle point is estimated (midpoint). **b** Looking for the entry scenario: left and right key locations are calculated by projecting the nearest point of the clusters left and right at *d* distance, respectively. If both key locations (left and right) are found, a middle point is estimated (midpoint)

As presented previously, the key locations *Kloc* are characterized by type p poses with orientation $\vec{v}$, which allows the necessary information to describe the closest rows to be contained in this representation.

Figure 3 shows an example of the estimation of entry key locations ($Kloc_{en}$) when the robot is in the *headland* (see Fig. 3b) and exit key locations ($Kloc_{ex}$) when the robot is in the *lane* (see Fig. 3a). As stated previously, to simplify the problem, $z_p = z_r$ is considered, assuming that the effect of the slope of the field on the key location estimation can be ignored and considering that the field of view of the perception system is narrow (only a few meters). To carry out the estimation, the following steps are performed: (1) the laser points (Lp) are projected into the Cartesian co-ordinate system $F_f'(x_f', y_f')$, (2) the line equations of both the left and right sides (y_l and y_r, respectively) are extracted by linear regression, and (3) based on these equations, $Kloc^l_{ex}$, $Kloc^r_{ex}$, $Kloc^l_{en}$, and $Kloc^r_{en}$ are calculated (see Fig. 3).

To calculate these key locations, a point to a user-defined distance d perpendicular to the orientation of the analyzed cluster is projected (see Fig. 3). In this particular example, distance d represents half of the inter-row distance or the desired distance to navigate in the alleys and is directly related to the tool/implement width. In general, distance d represents half the width of the lane. In the case of the exit points ($Kloc_{ex}$), the projection is performed with respect to the furthest point in the respective cluster at the front of the robot.

For the entry points ($Kloc_{en}$), the projection is performed with respect to the closest point in the respective cluster at the front of the robot. Distance d is first defined by the user and then updated while the algorithm is performing the detection to better estimate the row width. To update d, the Euclidean distance between the centroid of both left and right clusters closest to the current robot position is calculated. When the algorithm is reinitialized, i.e., a state transition occurs, distance d is reset to the user-defined value because all lanes cannot have exactly the same width.

For every entry and exit key location, if left and right are found, a key location $Kloc^m\left(p_m, \vec{v}_m\right)$ is set at the midpoint between both (i.e., *lane_end* key location type). The position p_m and orientation $\vec{v}_m$ of this middle point are given by:

$$p_m = \left(\frac{x_l + x_r}{2}, \frac{y_l + y_r}{2}, z_r\right) \tag{2}$$

$$\vec{v}_m = k \cdot \left(\vec{v}_l + \vec{v}_r\right), k = 1/2 \tag{3}$$

In some cases, particularly when gaps in the rows are found, both key locations ($Kloc^l$ and $Kloc^r$) may be distant from each other (when this distance is greater than half the distance of the field of view of the perception system). In this scenario, the furthest point (in the case of entry points) or the closest point (in the case of exit points) are discarded.

The algorithm presented constantly looks for all possible types of key locations independently of the current location of the robot on the topological map. A history of the estimated key locations is maintained, and the key locations that are no longer in the robot vicinity or do not meet the field model are discarded. Other conditions to discard a key location are the following: (i) the key location is behind the robot, (ii) the key location is far away, i.e., a distance greater than the length of the headland, (iii) the historical key locations do not correspond in orientation or alignment with the new key location (an error of approximately $\pm 10°$ or ± 0.4 m, respectively), and (iv) in the case of an exit type of key location, the new key location does not correspond in orientation or alignment with the robot. Therefore, the key locations that pass this scrutiny are named *valid key locations* ($VKloc_t$) and are classified as entry ($t = en$) or exit ($t = ex$). Only the key locations that are more likely to represent a state transition appear in this list. As previously stated, this set of key locations contains the required information for describing the crop rows. This information is used to carry out the semantic estimation of the current robot location on the topological map.

Current state estimation

To carry out the instantaneous estimated robot location *Est* based on a semantic analysis, two indices are calculated: P_index_t and A_index_t, where t is the type of the key location (*en* or *ex*). P_index_t corresponds to the Pearson correlation coefficient (PCC) between two random vectors (Benesty et al. 2009) and considers the same types of *valid key locations* in terms of the position (p_t) with respect to the field co-ordinate frame F_f and is given by

$$P\_index_t = corr\left(X_t\right), X_t = \left\{p_t\right\} \in VKloc_t \tag{4}$$

A_index_m corresponds to the correlation coefficient (PCC) between the robot orientation $\vec{h}$ and the orientation ($\vec{v}_t$) of the last *valid key location* identified, given by

$$A\_index_t = corr\left(Y_t, \vec{h}\right), Y_t = last\left(\left\{\vec{v}_t\right\} \in VKloc_t\right), \vec{h} \in Rob\left(r, \vec{h}\right) \tag{5}$$

For every considered working place (ignoring the *gate* in this case study), an index that determines how likely the robot is located in the different places is calculated, denoted as c_index_m, where $m \in M$ and M is a subset of N ($M \subseteq N$), which does not include the state *unknown*; see Eq. (1). Therefore, the calculation of each index is presented in Eq. (6) and can take values between 0 and 1:

$$\begin{cases} c\_index_{headland} = \frac{P\_index_{en}}{A\_index_{en}} \times cluster\_idx & if\ left \cup rigth\ alley\_idx = 0\ lane\_idx = 1 \\ c\_index_{alley} = \frac{A\_index_{ex}}{cluster\_idx} \times alley\_idx, & else \\ c\_index_{lane} = \frac{A\_index_{ex}}{cluster\_idx} \times lane\_idx & alley\_idx = 1\ lane\_idx = 0 \end{cases} \tag{6}$$

where $cluster\_idx$ is defined as follows:

$$cluster\_idx = abs\left(B_{dist} - F_{dist}\right) / euc\_dis(FC, BC) \tag{7}$$

Let us consider Fig. 4. Equation (7) expresses how misaligned the robot is with respect to the nearest cluster, calculated by projecting forward *FR* and backward *BR* points at a user-defined distance L in the Cartesian reference frame $F_f'\left(x_f', y_f'\right)$, and the nearby cluster points are located, designated *FC* and *BC,* respectively (see Fig. 4), referenced in the Cartesian reference frame F_f'. Then, the orthogonal distances B_{dist} and F_{dist} between the projected points (*FR* and *BR*) and the nearest cluster points (*FC* and *BC*) are calculated, where euc_dis represents the Euclidean distance between two points $p' \in \mathbb{R}^2$.

Therefore, these indices ($c\_index_{headland}$, c_index_{alley}, and c_index_{lane}) express the probability that the robot is located in the topological map, where the likelihood that the classification output is $z_k \in M$ given the actual class is $m_k \in M$ is presented in Eq. (8):

$$P\left(z_k | m_k\right) = \eta \sum_m c\_index_m \tag{8}$$

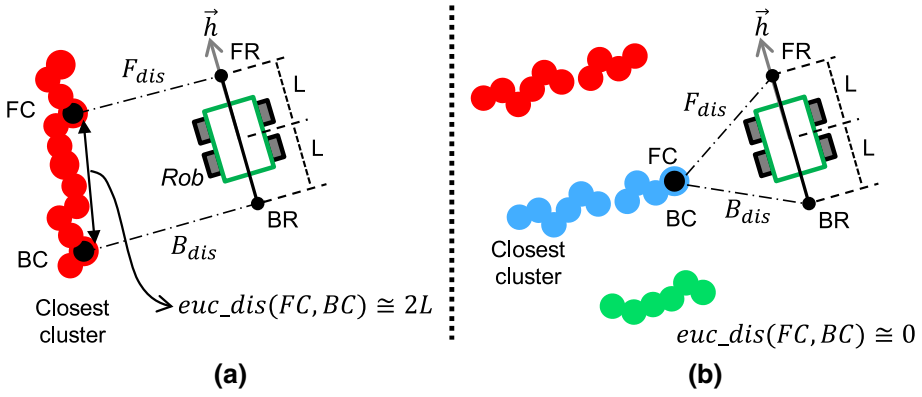

Fig. 4 Example of *cluster_idx* calculation. **a** Situation when the robot is in the lane. **b** Situation when the robot is in the headland

Consequently, Eq. (8) represents the observation model at instant k, where η is a normalizing constant. Subsequently, to estimate the instantaneous state Est_k of the robot on the topological map, the most likely state is taken, given by:

$$Est_k = n_k \leftarrow \max\left(P\left(z_k|m_k\right)\right) \tag{9}$$

where $n_k \in N$ as stated in Eq. (1). If the difference between two indices c_index_m is not sufficient to define a clear maximum (minimum difference of approximately 0.05), the instantaneous state estimation Est_k is defined as *unknown*.

The instantaneous observations do not consider past estimations and the action applied to the system when determining the current state. Given that the field model only allows certain transitions and that only the transitions can occur when a motion is applied, a filtering process is necessary to improve the robustness of the estimation. Therefore, to take advantage of these dependencies, the same concept as the one presented in Rottmann et al. (2005) and implemented in Weiss and Biber (2010) is used. It applies a Hidden Markov Model (HMM), which maintains the current estimated state based on the possible allowed transitions. This method requires an observation model, introduced in Eq. (8), and a transition model $P\left(m_k|m_{k-1}, u_{k-1}\right)$, which corresponds to the probability that the system moves from state m_{k-1} to state m_k by executing an action u_{k-1}, where $u \in U = \{motion, no-motion\}$. Therefore, the posterior probability $PE\left(m_k\right)$ of the state $m_k \in M$ given a previous observation, an instantaneous observation and allowed transitions is given by:

$$PE\left(m_k\right) = P\left(z_k|m_k\right) \sum_{m_{k-1}} P\left(m_k|m_{k-1}, u_{k-1}\right) PE\left(m_{k-1}\right) \tag{10}$$

The transition model only considers the probability of staying or changing state given a motion, since the probability of changing state given no-motion is zero. The transition model is usually built by simulating the working area, including an adequate representation of the dimensions of each of the different places. Then, the robot is randomly placed (including position and orientation) in the simulated working area, and an action is sent (in this case, a motion). This procedure must be performed with a representative number of samples to record the initial and posterior state. By following this procedure, the relationship between the state transition probabilities is directly proportional to the distribution of the working area, the size of each place, and the boundaries between places. Since this problem is similar to the one presented by Weiss and Biber (2010), it is not necessary to carry out this simulation again. It suffices to reuse its state transition probability and to consider the specificities of the model presented.

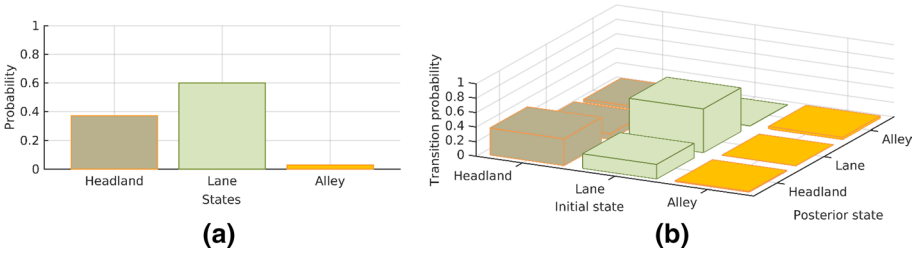

Fig. 5 Simplification of the state probabilities extracted from Weiss and Biber (2010). **a** Average probability distribution of the states studied. **b** State transition probabilities given a motion (from initial to posterior)

Therefore, Fig. 5 presents the simplification of the state probabilities extracted from Weiss and Biber (2010). Indeed, in the presented case, the problem is simplified because, instead of having six (6) states (open field, row, row start, row end, row gap and row side), the model presented only considers the three (3) main places for the topological map: headland (open field), lane (row) and alley (row side) (see Fig. 1a). The row start and row end disappear and become part of the main places, which are topological ambiguities. This ambiguity problem can be easily solved in the current case because the topological map presented includes a metric representation, which allows a precise estimation of the transition point to be estimated. This specificity makes this simplification possible, which is not the case if only a semantic analysis is used. This fact highlights the interest in using a hybrid topological map. In terms of navigation, the identification of gaps in the row does not make an important contribution, since the robot should continue to maintain the same behavior and not deviate toward the gap. In this sense, the gap (which is a situation commonly found in the field) is part of the lane in the presented model. Therefore, the current state is estimated by analyzing the posterior probability $PE(m_k)$ in Eq. (9).

In addition to the differences mentioned above with respect to work presented by Weiss and Biber (2010), it is worth highlighting the use of metric information so that the algorithm presented confers a significant improvement in terms of precise identification of state transitions between main working areas. Furthermore, the algorithm presented in this work uses a perception system capable of identifying and classifying different crop types, which allows one field to be isolated from another nearby field.

Transition evaluation

Of the set of *valid key locations* ($VKloc_t$), the last estimated locations are extracted for both entry and exit types. These key locations represent the possible locations of the transition between working areas on the metric map. Based on the current estimated state and the relationship between these points and the position of the robot, a transition is verified. Table 1 presents the truth table to verify a transition in the topological map, integrating the semantic state estimation and the metric map. A key location can also be considered reached if the point is passed by the robot.

Table 1 Truth table for evaluating the state transitions considering the current state estimated (Est_k), the last state (Est_{k-1}) and the relationship in terms of the distance between the last identified *valid key locations* (key loc.) and the robot position (*Rob*)

Current state	Headland			Lane/alley		
	Key loc					
Last state	Entry	Exit	Output	Entry	Exit	Output
Headland/unknown	Reached	–	True	Reached	–	True
	Far	–	False	Far	–	False
Lane/alley	Reached	Far	False	–	Reached	True
		Reached	True	–	Far	False
	Far	–	False			
Reached: $euc\_dis(p_t, r) \leq 0.01m$				Far: $euc\_dis(p_t, r) > 0.01m$		

Guidance control and map update

Given the current state estimation and the *valid key location*, the navigation strategy can be adjusted to meet the current objective provided by the topological map and the general mission. The identification of the *valid key locations* allows estimating the accurate position where a state transition occurs. The navigation strategy then relies on the search for possible lane candidates that require treatment (traversed). Therefore, each time a state transition is identified, the last *valid key location* is stored in an absolute metric map, referenced as O_f (see Fig. 1a). One possibility for building this map is to use a global localization system if it is available (i.e., GNSS), even if the uncertainty of this system does not allow proper and accurate navigation. Indeed, this map can be used to give the robotic system an idea of the necessary location and orientation to find a specific node within the hybrid topological map. Of course, once close to that location, the approach presented above must be used for precise navigation.

Materials and methods

Robotic platforms

The evaluation of the proposed methodology was carried out using two different robotic platforms. The first robot, the Oz robot (see Fig. 6b), is commercialized by Naïo Technologies for mechanical weed control, focused mainly on market-garden crops (Naïo 2020). This company lent the Oz robot to the Laboratory for Analysis and Architecture of Systems (LAAS) in the context of the abovementioned project called DESHERB'EUR. The second robot, Aircobot (see Fig. 6a), belongs to LAAS (Futterlieb et al. 2014). It was developed by a French company named Sterela (STERELA 2020) and is designed for general outdoor navigation. It has a similar mechanical structure to the Oz platform and similar sensor configuration. This second platform, higher and larger than the first one, was used to evaluate the methodology presented in vineyards. Oz's size is not adapted to this environment because of the low laser clearance

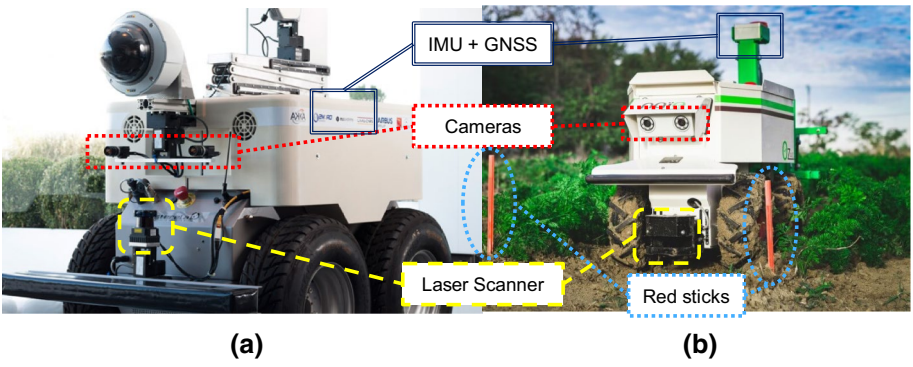

Fig. 6 Robotic platforms and sensor configurations. **a** Aircobot platform. **b** Oz platform. Red sticks are indicated in dashed (blue) circles

to the ground and high weeds that occlude the base of the trunks. Both platforms are equipped with the following sensors:

1. An RGB camera system, consisting of two cameras (left and right), with a vision range of approximately 107° for the Oz platform and 114° for the Aircobot platform with a resolution of 752×480 square pixels for both platforms.
2. A laser scanner that allows the surroundings to be perceived with a 270° vision range at a maximum distance of approximately 3 m for Oz and 5 m for Aircobot.
3. An inertial measurement system (IMU) and wheel odometry
4. A GNSS system with a variable resolution ranging between 0.6 and 5 m, depending on environmental conditions (clouds, branches, buildings, etc.).

Both the camera system and laser scanner are located at the front of the robots (see Fig. 6). Note that the RGB cameras were not used as a stereo vision system, given that the agricultural workspace environment introduces uncertainties due to the variability of illumination. However, the use of both monocular cameras allowed the field of view to be increased. Moreover, to attain depth on the perception system, the laser scanner data were merged with both cameras' outputs. Finally, the GNSS system was used to build the absolute map, which, as presented above, may be helpful for future navigation. However, given its poor and variant precision, navigation cannot rely only on this system.

To evaluate the proposed methodology, data were collected from real field operations to be analyzed off-line. At the time of data capture, the Oz platform was operating automatically: the robot was manually placed at the entrance of a lane or an alley (depending on the field), the artificial landmarks (red sticks) were placed as required, and the autonomous operation was activated. The robot worked at its maximum operating speed (0.36 m/s), and the laser, camera, GNSS and odometry (including IMU) data were stored. Moreover, the internal logs for building the ground truth were also recorded. Navigation for the Aircobot platform was performed manually (by remote control), simulating the behavior of a robot in this type of environment at an approximate speed ranging between 0.4 and 0.6 m/s.

Fig. 7 Samples of the datasets used for validation. From left to right and top to bottom: Broccoli 1 and 2; Cabbage 1, 2 and 3; Leeks 1 and 2; Vineyard 1

Table 2 Description of the datasets used for validation

Dataset name	Crop type	Location	Duration (s)	Images (left and right)	Laser scans samples	Data sample rate (Hz)	Field size (ha)
B1	Broccoli	Roquesérière	701	4172×2	4654	6	0.024
B2	Broccoli	Roquesérière	514	3006×2	3326	6	0.024
C1	Cabbage	Roquesérière	441	2707×2	2800	6	0.024
C2	Cabbage	Roquesérière	558	3094×2	3438	6	0.024
C3	Cabbage	Roquesérière	334	2009×2	2377	6	0.024
L1	Leeks	Escalquens	1259	8191×2	8371	6	0.0384
L2	Leeks	Roquesérière	92	602×2	615	6	0.0096
V1	Grapevine	Lisle-sur-Tarn	4179	29,252×2	41,800	10	0.798

Dataset profile

The proposed approach was tested and validated off-line with seven various datasets acquired from the Oz platform, recorded in real working conditions (see Fig. 7). Six of them were collected on the same farm located at Roquesérière (Haute-Garonne, France, 43.722° N 1.636° E) on two different days for three different types of crops (cabbage, broccoli, leeks). The remaining leek dataset was obtained in a greenhouse in Escalquens (Haute-Garonne, France, 43.515° N 1.572° E). In addition, an eighth experimental campaign was conducted using the Aircobot platform. The data were collected in an experimental vineyard in Lisle-sur-Tarn (Tarn, France, 43.841° N 1.851° E), characterized by its driving style (Guyot trellised) (IFV 2020). Thus, these datasets were acquired during various real situations: high changes in lighting conditions (different times of the day), diverse crops, locations, etc. Note that each dataset is made of synchronous raw images, as well as the laser point cloud referenced with the odometry and the GNSS system. Table 2 presents detailed information on the datasets used to validate the algorithm.

Ground truth definition

To define the ground truth required to evaluate the performance of the location system presented, two methods were applied depending on the robotic platform considered:

Oz platform

Artificial landmarks (red sticks, see Fig. 6b) were positioned in the field at the beginning and end of the rows. These landmarks are necessary requirements for the commercial platform to operate autonomously. The identification of the moment when the robot crossed these marks was recorded in an internal log, which was stored together with the other

sensors' information. The identification process was conducted very accurately by a vision-based onboard system built by Naïo, thus allowing us to define the ground truth. However, it must be noted that because of the crop considered, these landmarks were generally fixed slightly before the end of the row, inducing a small error of approximately ± 0.1 m in the definition of the ground truth. Nonetheless, this remains accurate enough for the purpose of the study.

Aircobot platform

A similar approach was followed, and artificial landmarks were created. They consisted of small wooden posts (0.02×0.02 m) covered with reflective material. This reflective material generated a peak of intensity in the laser readings, making it easier to identify the landmark position with respect to the robot. These posts were positioned next to the initial and final vine trunks on each row. Thus, in this case, they were positioned at the correct place to define the ground truth. Nonetheless, some metallic and wooden posts that belong to the trellis were found on the edges of the row. To consider them for navigation, the CNN was also trained to detect these features separately from the vine trunk. However, in the semantic analysis, laser points that belong to these posts may be labeled target crops. Therefore, an error of approximately ± 0.04 m in the ground truth definition was expected.

Results and discussion

Crop identification

As stated previously, to evaluate the selected network, YOLOv3 (Redmon and Farhadi 2018) and correctly classify the crops, a comparative study was performed with RetinaNet (Lin et al. 2017), another common network for object classification. B oth architectures and their variants were trained to identify an important variety of market-garden crops and later to identify the trunk of a grapevine. The training was performed using approximately 17,900 samples, of which 4/5 of the data were used for training and 1/5 for testing (see Table 3). The samples were extracted from the onboard cameras of the robotic platforms used for validation (Oz and Aircobot), performing real operations in different seasons, farms, lighting conditions, etc. Each sample was annotated by a hand-drawn bounding box, and each annotation consisted of an image where only the desired crop appeared. Given the similarity of the leaf shape and color, the CNNs were trained to identify leeks and onions as the same crop type. Moreover, cabbage and broccoli share the same statistical analysis, although the network can differentiate them as diverse crops.

Table 3 presents the results in terms of detection capacity and processing time for the different CNNs considered. A similar performance was obtained on the networks evaluated for all the configurations in terms of average precision (AP) and mean average precision (MAP). A threshold of approximately 0.5 of the intersection over union (IoU) was used for performance analysis. This means that it was defined as a positive identification if the estimated bounding box intersected with the annotated bounding box at least 50%. YOLOv3 288×416 was finally chosen for the methodology presented because it offers better results in terms of performance P (on a CPU and a GPU), as shown in Table 3. This short processing time allows its implementation on small robotic platforms such as the Oz robot

Table 3 Comparison of two CNNs trained for crop detection

CNN	AP-.50				MAP-.50			P (ms)	
	Cabbage & Broccoli	Rape-seed	Green lettuce	Red lettuce	Onion/leek	Grapevine		i7 6600U	GTX 1080
YOLOv3 416×416	0.871	0.676	0.966	0.957	0.499	–	0.794	2086.8	79.5
YOLOv3 288×416	0.860	0.694	0.968	0.964	0.4825	0.410	0.794	1460.7	56.9
RetinaNet-Resnet101 480×752	0.795	0.681	0.958	0.962	0.425	–	0.764	4451.2	62.5
RetinaNet-Resnet152 480×752	0.86	0.73	0.966	0.958	0.488	–	0.8	6124.8	72.4
Total samples	1301	1135	8258	1808	1444	3900			

Detection is considered true positive (TP) if the intersection over union (IoU) between ground truth and detection is greater than 50%

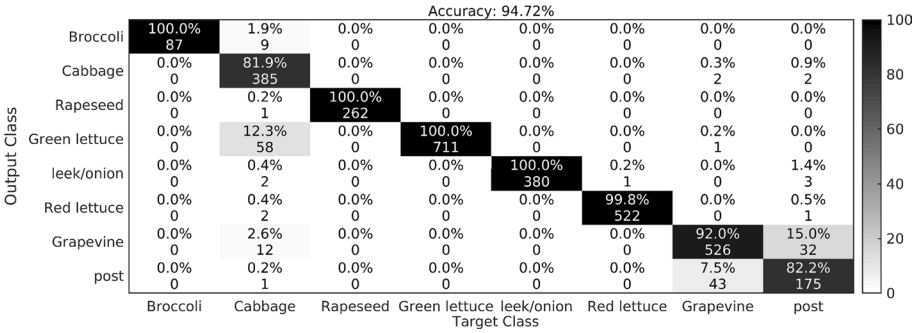

Fig. 8 Confusion matrix of the classification results of the trained crops for YOLOv3 (288×416). For each crop, true positive (the first diagonal) is presented, while misclassifications correspond to the rest of the cells

without requiring installing any extra or expensive hardware. Table 3 also shows that the average precisions of specific crops such as onion/leek and grapevine do not give similar results to the other trained crops. This result is due to the metrics that were used to measure the precision of the CNNs and to the type of crop. Indeed, it is a real challenge to make annotations of each of the abovementioned crops present in the images. Moreover, as it was decided to use the same analysis methodology as is commonly done in the scientific community, false positives were considered. Finally, during the annotation process, many crops were ignored because of their remoteness and relative size. Therefore, the analysis process yielded a considerable number of false positives, which reduced the precision for these specific crops in the statistical analysis presented in Table 3. In any case, this did not represent a problem with the proposed approach, since the crop classification was only performed by considering the field of view of the laser, which is shorter than the camera.

Figure 8 presents the confusion matrix obtained by analyzing the detection of each trained crop based on the dataset used for validation, where the x-axis shows the target class labels and the y-axis shows the output classes. Along the first diagonal are the correct classifications, whereas all the other entries show misclassifications. At the top of each cell, the percentage of correct classifications is shown, while below, the number of samples corresponding to these correct classifications is presented. The overall accuracy of the classification is 94.72%. Among the misclassifications, there are two situations that can be highlighted (see light grey cells):

1. The false classification of cabbages identified as green lettuce can be caused since both crops can be similar depending on the growth stage in which they are found, in addition to the fact that the network was trained with more samples of green lettuce than cabbages, causing overtraining.
2. Misclassifications between posts and grapevine: this is mainly because one object can occlude the other. The posts are in the row and hold the cables where the vine is held.

The second remark is not critical since both detections were used to positive filter the laser points. Nevertheless, it would be interesting to have a greater precision in the classification, especially to know whether the key location at the exit of the lane was estimated based on a post or a grapevine. This is due to the style used to support the vineyard (Guyot trellised), where the last post requires a diagonal cable anchored to the ground. This cable

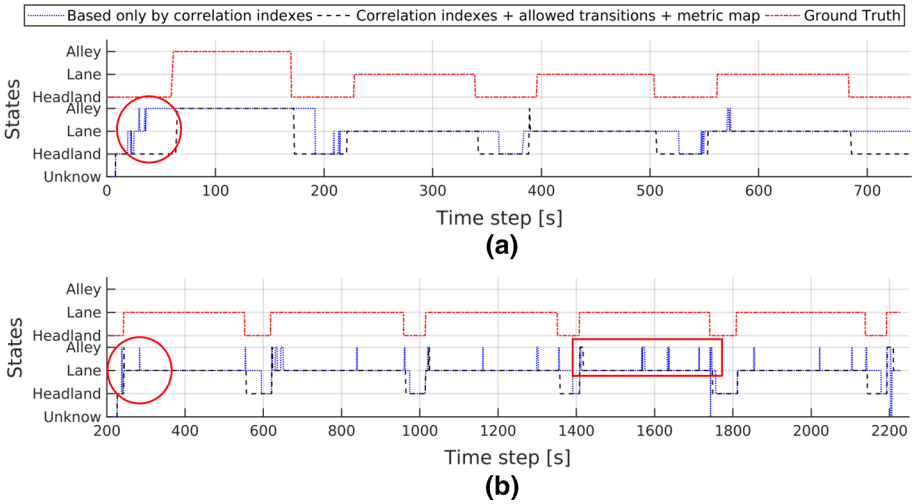

Fig. 9 Correlation index *c_idx_n* for each state studied. **a** Dataset B1. **b** A section of Dataset V1

creates an obstacle for the robot, and it is difficult to identify with the robot's sensory systems. In the space between the post and where the cable is anchored, there may or may not be a grapevine.

Semantic localization and key location results

Figure 9 presents the evolution of the correlation index *c_idx_n* defined by Eq. (4) for two experiments whose data were stored in two datasets denoted by B1 and V1. In the first dataset, the Oz platform was moving through one of four 50 m rows of a broccoli field (see

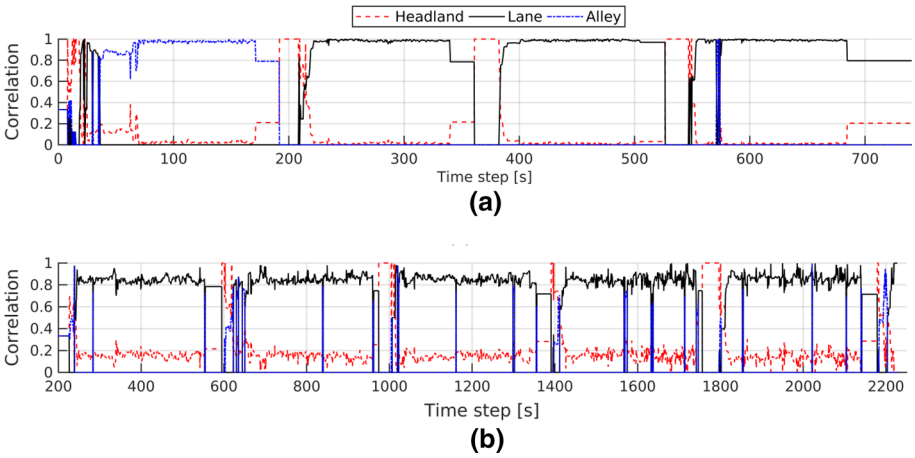

Fig. 10 State estimation *E* based on only the correlation index (blue dotted line) and on the previous estimation, the allowed transitions and the *valid key locations* (black dashed line). **a** Dataset B1. **b** A section of Dataset V1

Fig. 9a). In the second database, the Aircobot platform was traversing a part of a vineyard farm consisting of several fields with different sizes. More precisely, it was moving through five lanes with a length of approximately 140 m, collecting the data and computing the previously mentioned correlation index (see Fig. 9b). It is possible to observe that when the robot is in the lane or in the alley, the correlation indices for these places are significantly clear. The challenge lies in properly identifying the moment where a transition between states occurs, which in most cases is not obvious.

The instantaneous state of the robot on the topological map is computed by the maximum of the index given by Eq. (9) (see Fig. 9). The corresponding value obtained is presented as the blue line (dotted line) in Fig. 10a, b for the Oz and Aircobot platforms, respectively, and it is compared with the ground truth (red line, top). This estimation is sensitive to noise (see square Fig. 10b) since it is an instantaneous picture of the state of the robot and is mainly due to two factors:

1. Some failures to properly detect the crop may occur, given that the perception system may have false positives. This situation generally occurs when the robot is performing a U-turn in the headland, coming close to the crop row.
2. Some large gaps may appear. These gaps may occur when some crops are missing in the row or some crops occlude the camera and laser field of view, making row perception impossible. Although the first situation is rather common, especially in a vineyard, the second situation may occur when the robot becomes too close to a large crop with many leaves while navigating through the row. This problem occurs less frequently on the Aircobot platform because the embedded vision system has a relatively large baseline, which guarantees that at least one camera will be able to perceive the environment.

These different cases were filtered through the application of an HMM (black dashed line, see Fig. 10), which considers the permitted transitions and previous states. To further strengthen the estimation, the positions of the *valid key locations* with respect to the robots were also considered. Therefore, when the robot is moving through a lane (respectively, an alley) and a change in the state is estimated, the state is maintained in 'lane' (respectively, 'alley') if the vehicle is still far from a *valid key location* of the 'exit' type. The same procedure is used when the robot is in the headland and a *valid key location* of the type entry is searched.

The results obtained show that, in the case of the Oz platform, the localization, which relies on the sole correlation indices, is less accurate than the localization that considers this last information together with the allowed transitions and the metric map. This result is rather logical. Indeed, due to the robot size, turning radius, headland and inter-row dimensions, it is impossible to enter the next row without performing several forward and backward maneuvers (between five and seven). These maneuvers cause the robot to successively move in and out of the lane, which in turn generates false positive detections (see the red circle in Fig. 10a), as the transition evaluation assumes forward motion only. However, the integration of the transition within the localization process increases the accuracy and makes the system more robust to these perturbations.

In the case of the Aircobot platform, again, the interest in considering the metric map for validating the transitions clearly appears. Some sudden jumps can be observed in the localization relying on the sole correlation indices. They come from the real field, which does not exactly match the expected model presented in Fig. 1. Indeed, the field where the experimentation was conducted is planted diagonally, i.e., if an imaginary

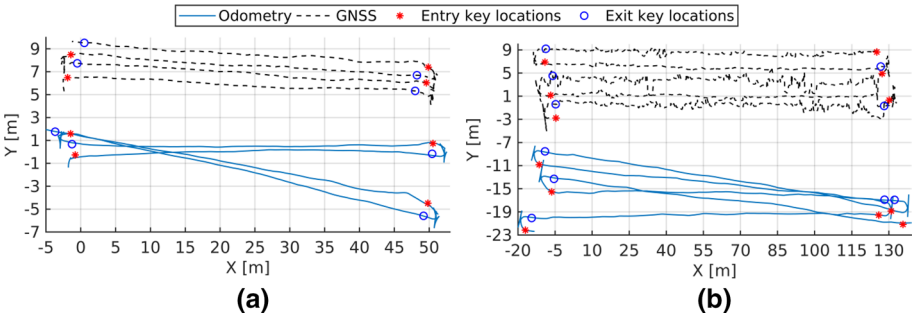

Fig. 11 Odometry of the robot and GNSS with key locations. **a** Dataset B1 with the Oz platform. Field dimensions: lane length: 50 m; lane width: 1 m; number of lanes: 4. **b** Dataset V1 with Aircobot platform Field dimensions: lane length: 140 m; lane width: 2.5 m; number of lanes: 5

line is drawn perpendicular to the crop orientation, the beginning of two consecutive crop rows does not coincide with that imaginary line. In such a case, each lane entry is first detected as an 'alley' and then updated to a 'lane'. From the point of view of the topological map, this is not handled as a state transition, since the model used does not allow such a transition. It is treated as an error in the initial estimation, and when the input data become consistent with a lane, i.e., when the 'lane' state is estimated for a sufficient number of epochs, the state is updated (see the red circle in Fig. 10b). This situation does not occur when exiting a lane because many epochs elapsed where a 'lane' state was estimated; thus, the model does not allow changing to an 'alley'. These types of situations are unusual in the field, although they may occur.

As an example of a possible use of the metric part of the topological map, Fig. 11 shows the odometry of the robots and the *valid key locations* for entry and exit on their respective tests. For the purpose of comparing the results, the *valid key locations* are also referenced with the GNSS sensor onboard the robots, which may allow creation of a map of the field to be used in the future. Figure 11 shows the GNSS data transformed into the same reference frame as the odometry to be able to observe and compare the

Table 4 State estimation and key location results for the datasets: B for broccoli; C for cabbage; L for leeks and V for vineyard

Set	MDE (m)				Orth_E (m)				Transitions		DSR (%)
	Entry	Exit	Mean	Var	Entry	Exit	Mean	Var	Entry	Exit	
B1	0.36	0.151	0.256	0.014	0.083	0.038	0.06	0.008	4	4	83
B2	0.829	0.533	0.681	0.008	0.3	0.187	0.244	0.007	4	4	88
C1	0.478	0.473	0.475	0.107	0.006	0.115	0.089	0.014	3	4	86
C2	0.655	1.153	0.904	0.077	0.052	0.137	0.095	0.022	3	4	92
C3	0.647	0.275	0.461	0.042	0.334	0.089	0.212	0.001	3	4	80
L1	0.921	0.868	0.895	0.070	0.148	0.458	0.301	0.004	8	7	81
L2	0.062	0.343	0.203	0.068	0.037	0.038	0.034	0.001	2	2	64
V1	0.446	0.434	0.44	0.058	0.174	0.122	0.148	0.013	19	14	95

For every key location type, the mean distance errors (MDE) of its position as well as the orthogonal projection errors (Orth_E) are calculated compared with the ground truth

different location sensors. The obtained absolute map will not guarantee accurate and safe navigation, given the GNSS accuracy limitations (± 0.6 m of positioning error for Oz and ± 1.2 m for Aircobot). This accuracy drops considerably in vineyards where the canopy reduces the GNSS signal quality.

Table 4 presents the detection success rate (DSR), in addition to the mean distance error (MDE) between the *valid key locations* and the ground truth. Moreover, the orthogonal distance error (Orth_E) is computed to identify how well a *valid key location* is estimated compared to the center of a lane. Even with the error introduced in the definition of the ground truth (see above), the obtained results demonstrate that it is possible to locate the row boundaries without the use of artificial landmarks. The obtained MDE is not very suitable for defining high-precision mapping procedures (millimeter level), although the computed Orth_E shows that the alignment of the key locations with respect to the ground truth was estimated with an average error of approximately 0.14 m. Although the number of available transitions may not be large enough to have a representative sample for some datasets, the results obtained confirm that the inclusion of a metric map alongside the topological map presents clear benefits for autonomous sensor-based navigation.

Figure 12 presents a statistical analysis of both the MDE and the Orth_E for estimating the key locations, comprising both entry and exit type for each error, including the 95% confidence interval. Although the confidence interval and the mean value of the MDE are relatively large, it is important to emphasize that the estimated key locations were aligned with respect to the ground truth, but placed either in front or behind the true location of state transition on the metric map. This can be better appreciated by analyzing the confidence interval and the mean value of Orth_E, which indicates that most of the key locations were estimated with an adequate alignment to the true point of change in state.

A video showing some of the results obtained is available by following this link: http://homepages.laas.fr/cadenat/PA19/PA19video.mp4.

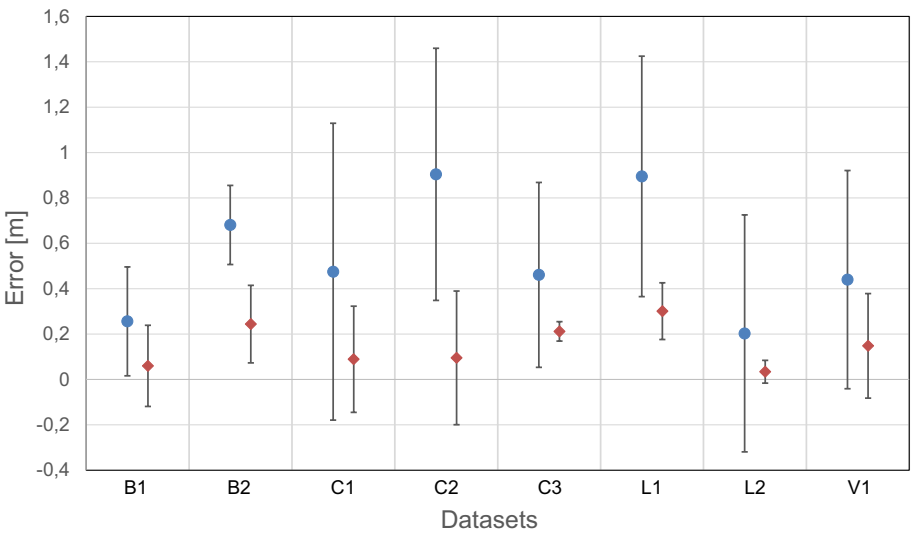

Fig. 12 Statistical analysis of the estimation errors of the key locations, including the 95% confidence interval. Mean distance error (circles in blue) and orthogonal distance error (diamond in red) for all datasets (Color figure online)

Future work will consist of coupling the approach presented to several sensor-based control strategies for obstacle avoidance, row following and U-turn. This coupling should improve the navigation and control capabilities of mobile agricultural robots, paving the way for low-cost and long-range navigation in agricultural environments.

Conclusion

This paper addresses the problem of map building for long-range autonomous navigation through agricultural fields. First, a model of the field is introduced and integrated into a hybrid topological map. This model considers the field as a part of a complete farm, which means that other fields with various crops can be found in the neighborhood. As a consequence, the navigation strategy requires a crop identification system to be efficient. Such a system was implemented by comparing different well-known object identifiers based on YOLO and RetinaNet-Resnet in terms of accuracy and processing time. These CNNs were trained with a considerable set of market-garden crops and grapevine trunks. The results obtained showed that YOLOv3 was the most efficient with respect to the previously mentioned performance indicators. This perception system also allowed the beginning and the end of the rows to be detected, avoiding the use of specific sticks or artificial landmarks and making the approach more generic. In addition, a methodology that allowed the robot to be located in the topological map was also developed and improved by integrating metric data to better detect the transitions between states. It benefits from the structure of the field to identify indistinguishable characteristics that allow the instantaneous robotic pose to be estimated. Moreover, the hybrid attribute of the map also makes it possible to improve the estimation of the state transitions by comparing the current robot pose on the metric map with the key locations, which represents possible points of state transition. The overall solution was validated using off-line data, recorded in real working conditions, from diverse fields with different crops. The results obtained allow evaluation of the approach scalability, which can be adapted to a considerable variety of crops and robotic platforms. The results demonstrated the efficiency and generality of the proposed approach to model and locate the robot in an agricultural field.

Acknowledgements The authors would like to thank colleagues from Naïo Technologies, Toulouse, for their participation in this work, through the collaborative project DESHERB'EUR funded by the program «Investment for the Future» of the French government. The authors would also like to thank the French Institute of Vine and Wine (*Institut Français de la Vigne et du Vin*), for allowing the use of their experimental fields for experimental tests.

References

Adão, T., Hruška, J., Pádua, L., Bessa, J., Peres, E., Morais, R., et al. (2017). Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing, 9*(11), 1110.

Ampatzidis, Y., De Bellis, L., & Luvisi, A. (2017). iPathology: Robotic applications and management of plants and plant diseases. *Sustainability, 9*(6), 1010.

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering, 149,* 94–111.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In I. Cohen, Y. Huang, J. Chen, & J. Benesty (Eds.), *Noise reduction in speech processing, springer topics in signal processing* (pp. 1–4). Berlin, Heidelberg, Germany: Springer.

Bergerman, M., Billingsley, J., Reid, J., & van Henten, E. (2016). Robotics in agriculture and forestry. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (pp. 1463–1492). Berlin Heidelberg, Germany: Springer.

Blanke, M., Blas, M. R., Hansen, S., Andersen, J. C., & Caponetti, F. (2012). Autonomous robot supervision using fault diagnosis and semantic mapping in an orchard. In G. G. Rigatos (Ed.), *Fault diagnosis in robotic and industrial systems* (pp. 1–22). California, USA: CreateSpace.

Blok, P. M., van Boheemen, K., van Evert, F. K., IJsselmuiden, J., & Kim, G. H. (2019). Robot navigation in orchards with localization based on Particle filter and Kalman filter. *Computers and Electronics in Agriculture, 157,* 261–269.

Bochtis, D. D., Sørensen, C. G., & Vougioukas, S. G. (2010). Path planning for in-field navigation-aiding of service units. *Computers and Electronics in Agriculture, 74*(1), 80–90.

Chen, K. H., & Tsai, W. H. (2000). Vision-based obstacle detection and avoidance for autonomous land vehicle navigation in outdoor roads. *Automation in Construction, 10*(1), 1–25.

Cherubini, A., Spindler, F., & Chaumette, F. (2014). Autonomous visual navigation and laser-based moving obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems, 15*(5), 2101–2110.

Comba, L., Gay, P., Piccarolo, P., & Ricauda Aimonino, D. (2010). Robotics and automation for crop management: Trends and perspective. In *Proceedings of the international conference on work safety and risk prevention in agro-food and forest systems* (pp. 471–478). Ragusa, Italy: Ragusa SHWA.

Durand-Petiteville, A., Le Flecher, E., Cadenat, V., Sentenac, T., & Vougioukas, S. G. (2017). Design of a sensor-based controller performing u-turn to navigate in orchards. In *Proceedings of the 14th international conference on informatics in control, automation and robotics (ICINCO)*. (Vol. 2, pp. 172–181). Setúbal, Portugal: SCITEPRESS.

Futterlieb, M., Cadenat, V., & Sentenac, T. (2014). A navigational framework combining visual servoing and spiral obstacle avoidance techniques. In *Proceedings of the 11th international conference on informatics in control, automation and robotics (ICINCO)* (Vol. 2, pp. 57–64). Setúbal, Portugal: SCITEPRESS.

García-Santillán, I., Guerrero, J. M., Montalvo, M., & Pajares, G. (2018). Curved and straight crop row detection by accumulation of green pixels from images in maize fields. *Precision Agriculture, 19*(1), 18–41.

Gonzalez-de-Santos, P., Ribeiro, A., Fernandez-Quintanilla, C., Lopez-Granados, F., Brandstoetter, M., Tomic, S., et al. (2017). Fleets of robots for environmentally-safe pest control in agriculture. *Precision Agriculture*, *18*(4), 574–614.

Hague, T., Marchant, J. A., & Tillett, N. D. (2000). Ground based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture*, *25*(1–2), 11–28.

Hamuda, E., Glavin, M., & Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture, 125,* 184–199.

IFV. (2020). Institut Francais de la Vigne et du Vin (French Institute of Vine and Wine). Retrieved September 23, 2020 from https://www.vignevin.com/.

Kanagasingham, S., Ekpanyapong, M., & Chaihan, R. (2020). Integrating machine vision-based row guidance with GPS and compass-based routing to achieve autonomous navigation for a rice field weeding robot. *Precision Agriculture, 21,* 831–855.

Kayacan, E., Kayacan, E., Ramon, H., & Saeys, W. (2015). Towards agrobots: Identification of the yaw dynamics and trajectory tracking of an autonomous tractor. *Computers and Electronics in Agriculture, 115,* 78–87.

Keskin, M., Sekerli, Y. E., & Kahraman, S. (2017). Performance of two low-cost GPS receivers for ground speed measurement under varying speed conditions. *Precision Agriculture, 18*(2), 264–277.

Kostavelis, I., & Gasteratos, A. (2015). Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems, 66,* 86–103.

Kuipers, B., & Byun, Y. T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems, 8*(1–2), 47–63.

Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition, 71,* 1–13.

Li, M., Imou, K., Wakabayashi, K., Tani, S., & Yokoyama, S. (2010). Position estimation method using artificial landmarks and omnidirectional vision. *Transactions of the ASABE, 53*(1), 297–303.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988). New York, USA: IEEE.

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., et al. (2015). Visual place recognition: A survey. *IEEE Transactions on Robotics, 32*(1), 1–19.

Malavazi, F. B., Guyonneau, R., Fasquel, J. B., Lagrange, S., & Mercier, F. (2018). LiDAR-only based navigation algorithm for an autonomous agricultural robot. *Computers and Electronics in Agriculture, 154,* 71–79.

Naïo. (2020). Robots agricoles Naïo Technologies (Agricultural robots Naïo Technologies). Retrieved November 23, 2020, from https://www.naio-technologies.com/.

Penizzotto, F., Slawinski, E., & Mut, V. (2015). Laser radar based autonomous mobile robot guidance system for olive groves navigation. *IEEE Latin America Transactions, 13*(5), 1303–1312.

Potena, C., Nardi, D., & Pretto, A. (2016). Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *Proceedings of the 14th international conference on intelligent autonomous systems* (pp. 105–121). Cham, Switzerland: Springer.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. Non-peer reviewed preprint at arXiv:1804.02767.

Rottmann, A., Mozos, Ó. M., Stachniss, C., & Burgard, W. (2005). Semantic place classification of indoor environments with mobile robots using boosting. In *Proceedings of the national conference on artificial intelligence (AAAI)* (pp. 1306–1311). California, USA: The AAAI Press.

Royakkers, L., & van Est, R. (2015). A literature review on new robotics: Automation from love to war. *International Journal of Social Robotics, 7*(5), 549–570.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS), 42*(3), 19.

Sharifi, M., & Chen, X. (2015). A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards. In *Proceedings of the 6th international conference on automation, robotics and applications (ICARA)* (pp. 251–255). New York, USA: IEEE.

Shamshiri, R. R., Weltzien, C., Hameed, I. A., Yule, I. J., Grift, T. E., Balasundram, S. K., et al. (2018). Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering, 11,* 1–14.

Shi, W., van de Zedde, R., Jiang, H., & Kootstra, G. (2019). Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering, 187,* 81–95.

STERELA. (2020). Société industrielle d'ingénierie et services (Industrial engineering and services company). Retrieved September 23, 2020 from http://www.sterela.fr/.

Thanpattranon, P., Ahamed, T., & Takigawa, T. (2016). Navigation of autonomous tractor for orchards and plantations using a laser range finder: Automatic control of trailer position with tractor. *Biosystems Engineering, 147,* 90–103.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence, 99*(1), 21–71.

Tu, C., Van Wyk, B. J., Djouani, K., Hamam, Y., & Du, S. (2014). An efficient crop row detection method for agriculture robots. In *Proceedings of the 7th international congress on image and signal processing* (pp. 655–659). New York, USA: IEEE

Vougioukas, S. G. (2019). Agricultural robotics. *Annual Review of Control, Robotics, and Autonomous Systems, 2,* 365–392.

Wäldchen, J., & Mäder, P. (2018). Plant species identification using computer vision techniques: A systematic literature review. *Archives of Computational Methods in Engineering, 25,* 507–543.

Weiss, U., & Biber, P. (2010). Semantic place classification and mapping for autonomous agricultural robots. In *Proceedings of IROS workshop on semantic mapping and autonomous, knowledge acquisition*. New York, USA: IEEE.

Xue, J., & Su, B. (2017). Significant remote sensing vegetation indices: A review of developments and applications. *Journal of Sensors, 2017*(1), 1–17.

Zhang, J., Maeta, S., Bergerman, M., & Singh, S. (2014). Mapping orchards for autonomous navigation. *Paper no. 141838567,* St Joseph, MI, USA: ASABE.

Zhao, Z.-Q., Zheng, P., Xu, S., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems, 30,* 3212–3232.