Computers and Electronics in Agriculture 151 (2018) 11–20

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Bounded memory probabilistic mapping of out-of-structure objects in fruit crops environments

Javier Gimenez*, Santiago Tosetti, Lucio Salinas, Ricardo Carelli

Instituto de Automática, Universidad Nacional de San Juan-CONICET, Argentina

ARTICLE INFO

Keywords:
Probabilistic mapping
Precision agriculture
Dynamic object
Recursive subsampling
Kernel estimators

ABSTRACT

Spatial awareness and memory are key factors for a robot to evolve in semi-structured and dynamic environments as those found in agriculture, and particularly in fruit crops where the trees are regularly distributed. This paper proposes a probabilistic method for mapping out-of-structure objects (weeds, workers, machines, fallen branches, etc.) using a Kernel density estimator. The methodology has theoretical and practical advantages over the well-known occupancy grid map estimator such as optimization of storage resources, online update, high resolution, and straightforward adaptability to dynamic environments. An example application would be a control scheme through which a robot is able to perform cautious navigation in areas with high probability of finding obstacles. Simulations and experiments show that large extensions can be online mapped with few data and high spatial resolution.

1. Introduction

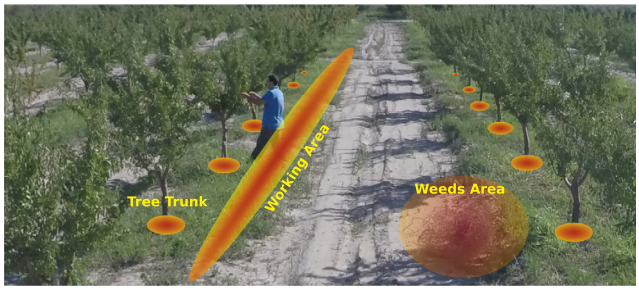
The United Nations proposed a 2030 agenda with 17 goals for sustainable development, placing food and agriculture as crucial pillars U. Nations. A profound change of the global food and agriculture system is needed to nourish today's 815 million hungry and the additional 2 billion people expected by 2050. One of the proposed objectives is to "Ensure sustainable consumption and production patterns", which points to do more and better with less. Precision agriculture is a modern farming practice that makes production more efficient by the proper application of inputs like water, fertilizer, pesticides, etc. at the correct time to the crop for increasing its productivity and maximizing its yields. Besides, precision agriculture provides farmers with a wealth of information to keep track of the farm, improve decision-making, ensure greater traceability, enhance marketing of farm products, improve lease arrangements and relationship with landlords, and enhance the inherent quality of farm products. A review of the motivations of implementing precision agriculture technologies is given in Pierpaoli et al. (2013).

In both developed and developing countries, the primary limiting factor in the development of agricultural industries is the manpower (Bechar and Vigneault, 2016), since it is the largest single cost-contributor in agriculture representing about 40% of the operational costs (Bechar and Eben-Chaime, 2014). The operational and socio-demographic factors that influence significantly in the adoption of precision agriculture technologies by German crop farmers are analyzed in Paustian and Theuvsen (2017). In the 20th century, technological progress in developed countries reduced the manpower for farming activities by a factor of 80 (Ceres et al., 1998). The transient nature of manpower in countries where wages are low reduces production capability and quality (Bechar and Vigneault, 2016). Furthermore, there are heavy manual tasks that cause injuries or chronic problems to workers (Perez-Ruiz et al., 2014). The enormous workforce force required for the different operations causes bottlenecks, downgrading productivity, reducing yield and increasing costs. Besides, problems such as aging of the workforce and shortage of rural workers contribute to the lack of manpower (Iida et al., 2013). This high manual labor requirement impedes cost reductions and increases the demand for robotics and automation (Bechar et al., 2007). Therefore, some human workers must be relocated to other sectors such as maintenance and programming of machines, supervision of tasks, or industrialization of primary agricultural goods (Autor, 2015). Statistics show that the agriculture labor is not lost but transformed (Employment Projections Program, 2017). A detailed analysis of the replacement of labor by machines is presented in Bechar and Vigneault (2016).

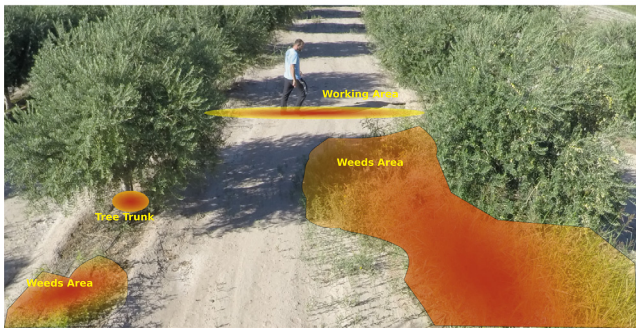
The use of robots enables the farmer to automate precision agriculture tasks (Yahya, 2018). There are already companies that offer robots that assist in agricultural tasks, such as Deepfield Robotics, Naïo Technologies, or Saga Robotics, to list some of them. The development of an agricultural robot must include the creation of sophisticated and intelligent algorithms for sensing, planning and controlling to cope with challenging, unstructured and dynamic agricultural environments

* Corresponding author.
E-mail addresses: jgimenez@inaut.unsj.edu.ar (J. Gimenez), stosetti@inaut.unsj.edu.ar (S. Tosetti), lsalinas@inaut.unsj.edu.ar (L. Salinas), rcarelli@inaut.unsj.edu.ar (R. Carelli).

https://doi.org/10.1016/j.compag.2018.05.018
Received 3 November 2017; Received in revised form 11 May 2018; Accepted 14 May 2018
0168-1699/ © 2018 Published by Elsevier B.V.



(a) Almond grove.



(b) Olive grove.

Fig. 1. Typical fruit grove environments with trees regularly located and workers performing a task. The working and weed areas are highlighted in order to identity regions of free movement and cautious navigation.

(Edan and Bechar, 1998). A review of recent research and developments in robotics for agricultural field applications and associated concepts, principles, gaps, and limitations can be found in Bechar and Vigneault (2016). One of the main limitations is the uncertainty accumulation during navigation, which can be reduced by including landmarks on the whole farm (such as QR codes in each tree) or by using prior information (such as an environment map or particular distributions of the trees), among others. For this reason, the mapping of partially structured agricultural environments is a valuable resource for precision agriculture. The regular geometry of the trees in an orchard (see Fig. 1) allows having prior precise information of their locations, which can be exploited by several automatic systems (Gimenez et al., 2015). Nevertheless, other objects, which are external to this regular structure, are also present in the agricultural environment. Among others, weeds, fallen branches, machinery, and rural workers are considered out-of-structure objects. These elements are also essential to be mapped because they can affect the robot navigation as well as the Human-Robot interaction (HRI) tasks. In such environments, the robot must be part of a more adaptive system, with the possibility to dynamically introduce new objects to the scene (Kaldestad et al., 2012).

Weeds and workers are part of any agricultural environment (see Fig. 1). Weeds affect the production by reducing crop yield and quality, delaying or interfering with harvesting, preventing water flow, etc. (Scursoni et al., 2011; Zimdahl, 2007). Tasks and other elements for weed control can be optimized by detecting and mapping the areas where weeds are more likely to grow (Torres-Sospedra and Nebot, 2014; Dammer, 2016; Panetta, 2015). On the other hand, the mapping of rural worker traffic allows generating appropriate control algorithms for HRI. Human capabilities of perception, thinking, and action are still unmatched in environments with anomalies and unforeseen events (Tervo and Koivo, 2014). Consequently, human and robot skills are complementary.

Dynamic environments are best represented with probabilistic maps in which areas with the highest probability of occupation are highlighted (see the working area in Fig. 1). This mapping procedure is often associated with the costly methods of occupancy grid maps in which the environment is regularly partitioned, and a probability is assigned to each grid cell (Thrun et al., 2005). In large-scale environments or when there is the need for high resolution, memory consumption of this methodology can become prohibitive, and even more if a 3D map is desired. The grid-based maps can be optimized using octrees, that allows to generate an original map with low resolution and refine each grid as their occupation probability increases (Hornung et al., 2013). The tree representation of the map reduces access times, memory consumption, and can also be used as a multi-resolution representation since it can be cut at any level to obtain a coarser subdivision (Hornung et al., 2013). However, these optimizations are not naturally designed to work in dynamic environments, and the incorporation of probabilities is not straightforward since it requires probabilistic models that are heuristically adjusted.

The main contribution of this paper is the development of a probabilistic method for mapping out-of-structure objects with the following properties: (i) It only requires storing the coordinates of an observation set detected outside the regular structure of the orchard; (ii) The observation set allows a nonparametric estimate of the unknown density function f of the out-of-structure objects through a Kernel estimator; (iii) It obtains a probabilistic map with high spatial resolution; (iv) It allows adapting the amount of stored information to the processing and storage capacities without compromising the map spatial resolution. The mapping procedure also allows incorporating new areas without increasing the required memory space. This is achieved by using a novel recursive subsampling methodology, which eliminates redundant non-informative data and reduces outliers. (v) The access times to the data can be optimized if a tree structure (like octrees) is generated, in which the observations are grouped according to the similarity of the decimal representations of their coordinates. (vi) It does not require constant updating of the probabilistic map, and the probability of observing an object at a specific point (and not the probability of finding an object within a grid cell) can be estimated online; (vii) It does not require: initialization, prior knowledge of the areas to be mapped, nor perform costly copy operations every time the map area is expanded; (viii) Free and unknown areas are not stored, and they are detected by the absence of points in spatial windows. (ix) It allows mapping dynamic environments by incorporating a forgetting factor. In this map, there are no regions without a significant probability of objects presence. (x) Kernel estimators are consistent and probabilistically optimal. Instead, the histogram estimator (or its generalization in grids) requires a cell size reduction (increasing the amount of storage memory required) to converge theoretically to f while increasing the sample size (Györfi et al., 2002). (xi) The estimates do not need the probabilistic modeling of the sensor, which generally contains heuristically adjusted parameters. (xii) It facilitates loop-closures in slam processes since original observations are stored instead of increasing data counters in each grill losing spatial information.

In addition, this paper presents an example application in which a robot uses this map to achieve cautious navigation by reducing its velocity in areas with high probability of finding rural workers. This navigation strategy includes an obstacle avoidance controller based on impedance. Fig. 1 presents two typical environments where the proposed system can be applied. In the first case, a worker performing a task on trees of the same line is observed. The points generated by these observations are incorporated into a database, which is kept bounded by using a recursive subsampling procedure. If the observation area is frequently occupied, then the algorithm marks it as a high traffic zone (working area in Fig. 1) and activates a cautious navigation mode. However, if the worker is never again observed in this region, the algorithm will forget these observations over time. Weeds are also detected and marked as areas with collision probability, but if they are removed, this area will be marked as a free area again. Fig. 1(b) presents a similar situation, in which a worker crosses the likely path of the robot. If this passage is commonly used for workers (or machines), the

algorithm assigns it a high collision probability and activates a cautions navigation mode in this area. In both cases, the regular distribution of the trees can be used to keep the mapping errors bounded. This mapping procedure should be seen as a complementary tool to be used by the robot while it performs its assigned task. The proposed methodology is not intended to replace the human, but to improve the Human-Robot Interaction.

The paper is organized as follows. Fundamental concepts of density estimation to develop this paper are formally defined in Section 2. Section 3 presents an online recursive subsampling methodology used to keep the points cloud size bounded, reducing non-informative observations and outliers. In Section 4, it is explained how to make the probabilistic map of the out-of-structure objects by using the Kernel estimator. An example application of probabilistic mapping is developed in Section 5. Simulations of the probabilistic mapping of weeds and rural workers are shown in Section 6. Experimental results are shown in Section 7, and finally, conclusions are given in Section 8.

2. Density estimation

The density of a random variable is a normalized function that takes more significant values in the most likely points to observe the variable. Density estimators seek to maximize some functional that depends on a finite sample $X_1, \ldots, X_n$ of the distribution. The most known optimality criterion is the maximum likelihood approach, which seeks the density f that maximizes the joint likelihood of all observations, that is,

$$\hat{f}_n = \underset{f}{\operatorname{argmax}} \prod_{i=1}^{n} f(X_i). \tag{1}$$

Density estimators are divided into parametric and nonparametric. On the one hand, parametric estimators assume that the density is characterized by a parameter set, like Gaussian or Gaussian mixture, wherein the parameter estimation is equivalent to the density estimation. These models are not the most advisable to characterize unstructured environments such as those found in this work. On the other hand, nonparametric estimators seek to optimize (1) assuming that f belongs to a class characterized by functional concepts such as smoothness or shape (Györfi et al., 2002). Among the most known nonparametric estimators, the histogram estimators (or its generalization based on grids) and the Kernel estimators are highlighted.

Density estimators based on grids are the most commonly used in the application field, in which the domain of f is discretized, and the observation probability on each grid cell is optimally estimated (according to (1)) by the proportion of observed points in the cell. In robotic mapping, octrees avoid the full data storage, which is the main shortcoming of fixed grid structures.This is achieved by mapping in low resolution and delaying the sub-grids initialization until the occupation probability of the grid becomes significant.

In this way, the extent of the mapped environment does not need to be known beforehand, and the map only contains grids that have been measured (Hornung et al., 2013).

A discretization of the environment can be avoided by storing range measurements directly. The occupied space in the environment is then modeled by the point clouds returned by range sensors. Kernel density estimator (KDE) is based on a finite data sample and can be viewed as a generalization of histogram density estimation with improved statistical properties. In short, the KDE performs the convolution between the sampling point cloud and a kernel in charge of weighting local observations.

Given a random sample $X_1, X_2, \ldots, X_n$ of a population, the 1D version of KDE is defined by

$$\hat{f}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$

where K is the kernel (a non-negative function that integrates to one and has zero mean), and $h > 0$ is the bandwidth. Several kernel functions are commonly used: uniform, triangular, biweight, triweight, Epanechnikov, normal, among others (Györfi et al., 2002). Intuitively, the smoothing parameter h is choosed as small as the data allow, however there is always a trade-off between the bias of the estimator and its variance. If the bandwidth is small (big), then an undersmoothed (oversmoothed) estimation with low (high) variance and high (low) bias is produced. Several optimality approaches are used to select the appropriate bandwidth (Györfi et al., 2002).

In this paper, out-of-structure objects are mapped in a 2D space, but generalization to higher dimensional spaces can be performed. Let $(X_1, X_2, \ldots, X_n) \in \mathbb{R}^2$ be an independent and identically distributed sample drawn from some distribution with an unknown and nonparametric density f. The bidimensional kernel density estimator of f is defined by (Botev et al., 2010)

$$\hat{f}_n(x) = \frac{1}{n|H|} \sum_{i=1}^{n} K(H^{-1}(x - X_i)), \tag{2}$$

where K is the kernel (a non-negative bidimentional function that integrates to one and has zero mean), and $H = \operatorname{diag}(h_x, h_y) > 0$ contains the bandwidths. The standard bivariate normal kernel is often used due to its convenient mathematical properties, this is,

$$K(x) = \frac{1}{2\pi} \exp\left\{-\frac{1}{2} x^T x\right\}.$$

Once the kernel function is selected, the estimator (2) becomes parametric, since it now results to be characterized by the parameters h_x and h_y. More details on how to optimally define the smoothing parameters of 2D-KDE are given in Wand and Jones (1993) and Botev et al. (2010).

If the probabilistic reconstruction of the whole map is desired, the Matlab function *kde2d* (Botev, 2007) can be used, which is a fast and accurate state-of-the-art bivariate kernel density estimator with diagonal bandwidth matrix. The two bandwidth parameters are optimally chosen without ever using a parametric model for the data or any "rules of thumb". Unlike many other procedures, this one is immune to accuracy failures in the estimation of multimodal densities with widely separated modes (Botev et al., 2010).

Instead, if the probability of detecting an out-of-structure object at a specific point x (and not the grid cell to which it belongs) is desired, the observations lying within a window centered in x should be selected and weighted by using (2). The bandwidths can be calculated online as indicated in Botev et al. (2010), or it can be used precalculated optimal values.

Note that it is not necessary to make any probabilistic calculation during data collection. Besides, the size of the data set can be adapted to the processing and storage capacities. To this, the database must be constantly subsampled, keeping it updated and bounded. However, the higher the storage capacity, the better the estimate of f, since $\lim_{n\to\infty} \hat{f}_n = f$.

3. Recursive subsampling

The number of observations continually increases throughout the data acquisition, and therefore, the probability computation can become a costly task if a mechanism for eliminating non-informative observations is not incorporated. Although most methodologies incorporate processes to detect and remove outliers, it is impossible to eliminate them altogether. On the other hand, repeated or redundant observations can appear if too much data are accumulated. For this reason, the data subsampling reduces outliers significantly, and in the same proportion, eliminates redundant observations.

Subsampling is the random process of reducing the sample size from K to $k < K$ (subsampling performed in one step), in which each observation remains in the subsample with probability k/K. Suppose now

that there is a storage capacity quota of C observations, which is lesser than the expected amount of data. Then, it is not feasible to wait for the experiment finishes for then performing a subsampling in one step. This requires to perform a subsampling, every time the number of stored observations N exceeds the quota C, in which each observation has a probability C/N of remaining in the subsample (recursive subsampling).

With a traditional recursive subsampling, in which observations are randomly deleted as new observations are acquired, the more recent observations have a higher probability to be in the subsample. Thus, this recursive subsampling is not equivalent to a subsampling performed in one step, and it tends to quickly forget old representative observations. Algorithm 1 produces a recursive subsampling equivalent to a subsampling performed in one step, in which all observations (regardless of when they are acquired) have the same probability of belonging to the point cloud. This purpose is fulfilled through an acceptance-rejection method of new observations.

At the beginning of the algorithm, the point cloud is empty ($N = 0$), and the probability of accepting new observations is $p = 1$. The algorithm incorporates all the acquired observations to the point cloud until the storage quota is exceeded $N > C$. When this happens, C observations are subsampled without replacement, and the remaining observations are deleted. Then, the probability of belonging to the point cloud is updated to $p = C/N$, and N is set to C. From this moment, each new observation must undergo an acceptance-rejection process equaling its probability to belong to the point cloud with the probability of the observations acquired previously. This probability of belonging p decreases every time a new subsampling is performed (steps 15–17 of the Algorithm). In this way, all observations (originals and news) have the same probability p of belonging to the point cloud. The **for** loop of the steps 7–13 can be efficiently performed in matrix form.

Algorithm 1. Subsampling algorithm

1: **Set** $N = 0$ ▷ size of the point cloud
2: **Set** $p = 1$ ▷ probability for the acceptance-rejection procedure
3: **for** $t = 1$: end **do** ▷ for each sampling period
4: Include the n_t new observations in the point cloud.
5: $N = N + n_t$
6: **if** $N > C$ **then** ▷ if the storage quota is exceeded
7: **for** $i = 1$: n_t **do** ▷ for each new observation
8: $u = \text{rand}(0, 1)$ ▷ random number with uniform distribution
9: **if** $u > p$ **then** ▷ acceptance-rejection procedure
10: i-th new observation is deleted
11: $N = N-1$
12: **end if**
13: **end for**
14: **if** $N > C$ **then** ▷ if the storage quota is still exceeded
15: a subsampling without replacement of size C is performed.
16: $p = pC/N$ ▷ update the acceptance-rejection probability
17: $N = C$
18: **end if**
19: **end if**
20: **end for**

Fig. 2. Google Maps image of an experimental olive grove at the National Institute of Agricultural Technology (INTA) located in San Juan, Argentina. This fruit grove has a regular geometry in which the olive trees spacing is 6 m (between rows) by 4 m (between trees).

A forgetting factor can be set according to the application by modifying the update of the acceptance probability of step 16. A possible update may be $p = p(C/N)^{\alpha}$, $\alpha > 0$.

4. Mapping out-of-structure objects

Consider a fruit grove with regular geometry, like the olive grove shown in Fig. 2. Let $\mathbf{m}$ be the set of actual locations of the trees. Any object outside of this regular structure is considered an out-of-structure object, which can be dynamic or static. Among dynamic objects are the farm workers and machineries, and among the static objects, it can be mentioned the weeds and complementary structures. In this paper, it is considered one of each type: agricultural workers and weeds.

Let $\mathbf{x}_t = [x_{t,x}, x_{t,y}, x_{t,\theta}]^T$ be the state variable of the vehicle at time t, where $(x_{t,x}, x_{t,y})$ represents its position and $x_{t,\theta}$ its orientation. In addition, it is assumed that the vehicle is equipped with a laser range sensor whose measurements can be summarized in a vector $\mathbf{z}_t = \{z_{t,i}: i = 1, ..., n_t\}$, where each $z_{t,i} = [z_{t,i,d}, z_{t,i,\theta}]$ contains the distance $z_{t,i,d}$ from the robot to the i-th obstacle, and the direction $z_{t,i,\theta}$ from which this obstacle is observed. This direction takes values between 0 (right side of the robot) and π (left side of the robot). Then,

$$\Omega_t = \left\{ \begin{bmatrix} x_{t,x} \\ x_{t,y} \end{bmatrix} + z_{t,i,d} \begin{bmatrix} \cos(z_{t,i,\theta} + x_{t,\theta} - \frac{\pi}{2}) \\ \sin(z_{t,i,\theta} + x_{t,\theta} - \frac{\pi}{2}) \end{bmatrix} : 1 \leqslant i \leqslant n_t \right\},$$

is the observed point cloud at time t (see Fig. 3 for more details).

Each point of these sets is considered a tree observation if its distance to the regular grid is less than $\delta > 0$, where the threshold δ is a design parameter. Then, each $\mathbf{z}_t$ can be partitioned into tree observations $\mathbf{z}_t^a$ and out-of-structure observations $\mathbf{z}_t^e$.

In this paper, the observation $\mathbf{z}_t^a$ is used to perform a classic SLAM to correct small differences between $\mathbf{m}$ and the actual regular structure, and then to improve the estimation of robot trajectory $\{\mathbf{x}_t\}$. $\mathbf{m}$ and $\{\mathbf{x}_t\}$ can be calculated with high accuracy if the SLAM algorithm is initialized with the regular map and the exact initial pose of the robot.

On the other hand, the point cloud of out-of-structure objects is built by using the vectors $\mathbf{x}_t$ and $\mathbf{z}_t^e$ for all t. This database is kept representative, updated and bounded by using Algorithm 1, which can be run online.

5. Application example of the probabilistic map

Human-Robot Interaction (HRI) in the agricultural environment will be a typical situation shortly, and therefore, it represents a topic of great research interest. In HRI, not only the human bt also the robot is

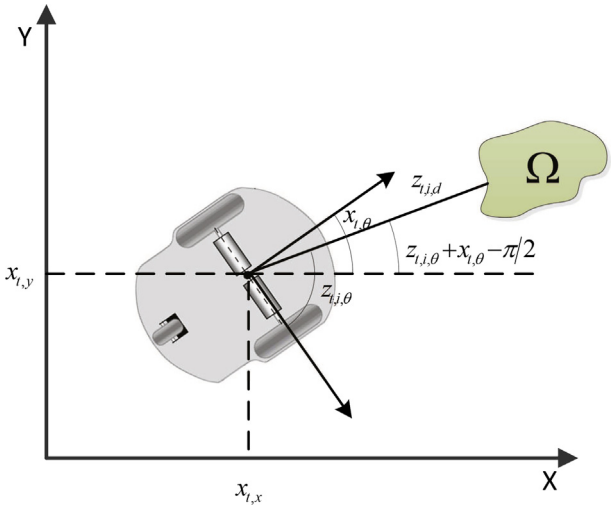


Fig. 3. Robot pose and variables based on laser measurements that characterize the relative position of the detected objects.

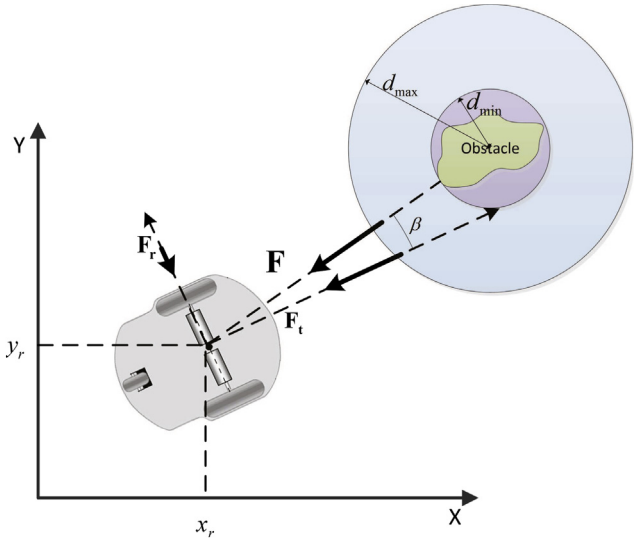


Fig. 5. Obstacle avoidance system. When an obstacle is detected, the robot is repelled by virtual forces ($\mathbf{F} = [\mathbf{F_r}, \mathbf{F_t}]$) preventing likely collisions. These forces depend on preset safety distance parameters (d_{min} and d_{max}) and the vehicle-obstacle relative orientation (β).

expected to adapt to its partner (Mitsunaga et al., 2008). Current research is aimed at achieving automated farms where machines and humans coexist in total sync, although this is a significant challenge that requires passing a countless number of technological and social barriers (Ellis et al., 1991; Bergerman et al., 2015).

A social robot (Dautenhahn and Billard, 1999) in agriculture must have socially-acceptable movements, i.e., their movements should be legible, natural, and should take into account the human comfort (Auat Cheein et al., 2015). Successful farm interaction depends strongly on the appropriateness, usefulness, and efficiency of the robot (Latif et al., 2009). A mobile robot that navigates between humans and other machines must follow a series of security and social norms. If the robot detects map zones with higher traffic and obstacle presence, then it should be able to navigate along these zones cautiously.

The statistical significance of the observations becomes relevant when the robot senses dynamic objects since erroneous maps can be generated if the total observation time is not long enough. If no workers are observed in some region of the crop, it does not mean that in this region is unlikely to find them. In the same sense, if a worker is observed in an atypical place, false areas with worker occupancy could be generated. The statistical theory states that the only way to reduce these uncertainties is by increasing the sample size, i.e., increasing the period in which the robot acquires data, and the times the robot passes over the same path. This concept can be compared to humans gained experience, because the more times a situation is faced, the better is the response to that situation. This procedure could be incorporated into agricultural robots to acquire additional information to perform its assigned task under an HRI paradigm.

An example application of the probabilistic map is presented in this Section. The aim of this example is not to prove the mapping abilities of the robot but to iteratively construct a dynamic map based on the information gathered in successive visits, to achieve an efficient, secure and socially acceptable navigation.

Consider an agricultural environment with regularly distributed fruit trees, and a robot equipped with a laser range sensor and a control system, as shown in Fig. 4. The semi-structured characteristic of the environment allows the algorithm to easily distinguish trees from out-of-structure observations, such as humans, machinery or any other obstacle that restrict the free transit areas (*Object classification* block in Fig. 4). Trees observations are used in the odometry correction (*SLAM* block in Fig. 4), while out-of-structure observations are incorporated in a database of 2D cartesian coordinates by using Algorithm 1 (*Point cloud update* block in Fig. 4).

Consider now that the agricultural mobile robot is performing some specific task. For simplicity, in this case it is assumed that the task is to follow a specific path across the crop, having into account the presence of obstacles. While performing its assigned task, the robot should be able to reduce its velocity in zones with high obstacle probability even if momentarily there are not out-of-structure objects in the area. This

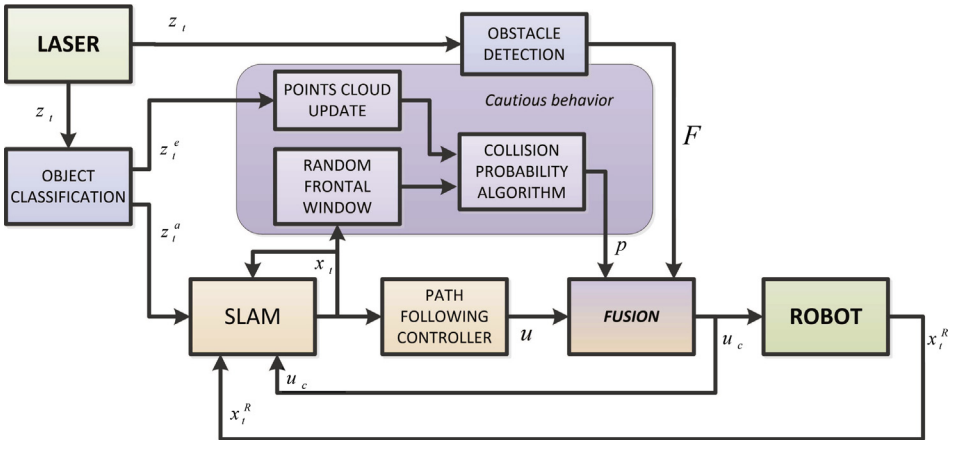
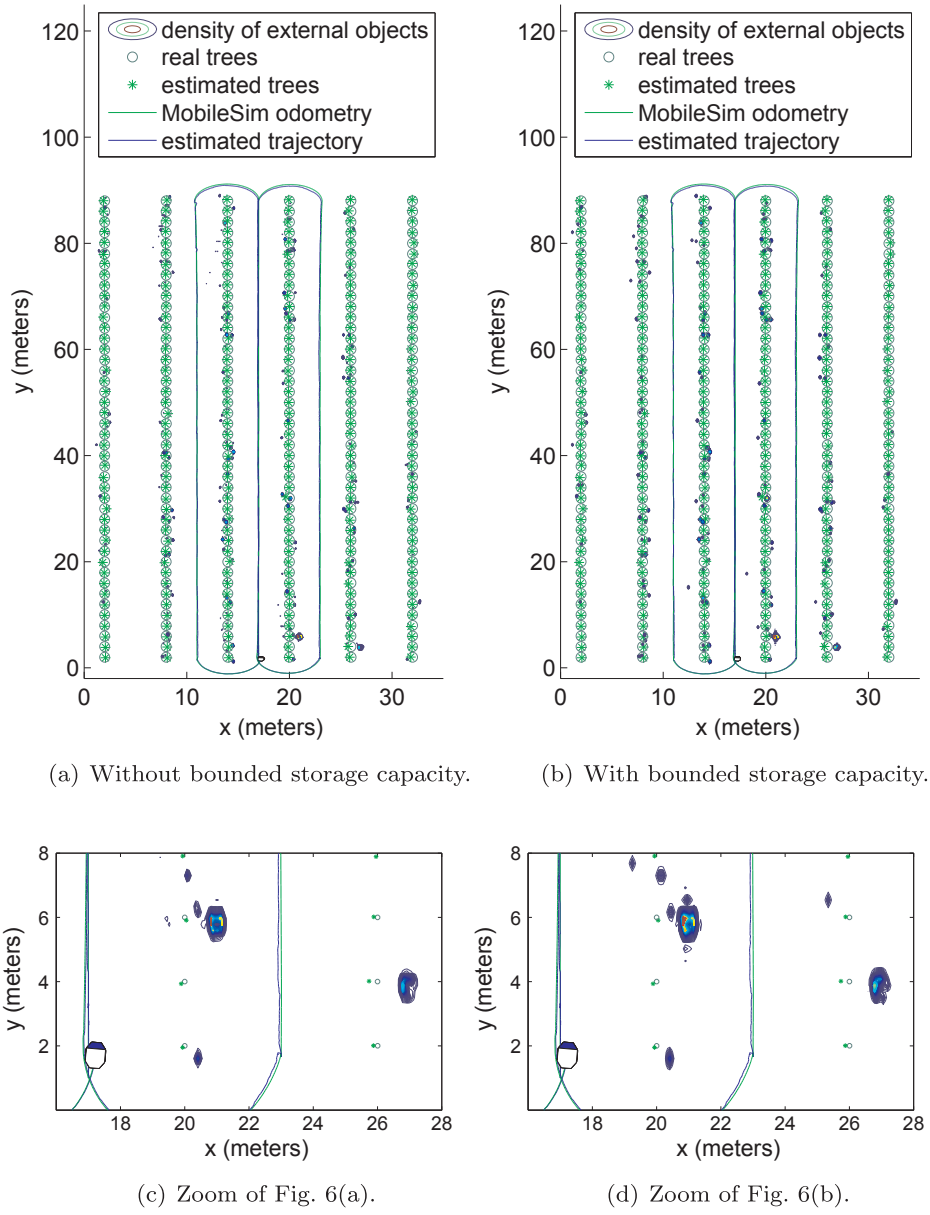


Fig. 4. Control system diagram for the example application. Laser measurements are classified as tree or as out-of-structure observations, and then assigned to SLAM (to reduce system errors) or to update the point cloud (on which the probabilistic map is based), respectively. The control action, given by the path following controller and modified by the obstacle avoiding system, is fused with the collision probability to generate a cautious navigation for the robot.



(a) Without bounded storage capacity. (b) With bounded storage capacity.

(c) Zoom of Fig. 6(a). (d) Zoom of Fig. 6(b).

Fig. 6. Outcomes of the first simulation. The mapping considering a storage bound takes up less memory and obtains maps similar to those based on the whole database.

deceleration is similar to the velocity reduction that a car driver performs while approaching a crossing street.

The probabilistic map is generated following Algorithm 2 (see the *cautious behavior* block in Fig. 4). Several controllers can be used to perform steps 5 and 6 of Algorithm 2 (see the *path following controller* and *obstacle detection* blocks in Fig. 4). Simulations and experiments presented in this paper use the path following controller developed in Andaluz et al. (2011), with a modification for obstacle avoidance based on impedance (Secchi and Mut, 2003).

Algorithm 2. Cautious behavior

1: **while** performing a normal navigation **do**
2: **define** a rectangular area in front of the robot;
3: **randomly sample** the area and calculate the transit probability at these points by using (2);
4: **estimate** the collision probability p by averaging these probabilities;
5: **calculate** the desired velocities by using a path following controller;
6: **modify** the control actions according to the requirement of the obstacle evader;
7: **multiply** the linear velocity by the reduction factor $1/(1 + k_c\sqrt{p})$, where $k_c > 0$ is a design constant.
8: **end while**

The path following control action is $\mathbf{u} = [\nu, \omega]^T$, where ν and ω are the linear and angular desired velocities. The detected obstacles produce a repulsion force $\mathbf{F}$, which is decomposed in a lateral component $\mathbf{F_r}$ and a longitudinal component $\mathbf{F_t}$ according to the robot-obstacle relative orientation (see Fig. 5).

The control action $\mathbf{u}$ and repulsion forces $\mathbf{F_t}$ and $\mathbf{F_r}$ are combined with the collision probability p to modify the velocity and orientation of the vehicle, as shown Fig. 4. The *fusion* block of Fig. 4 implements the equations

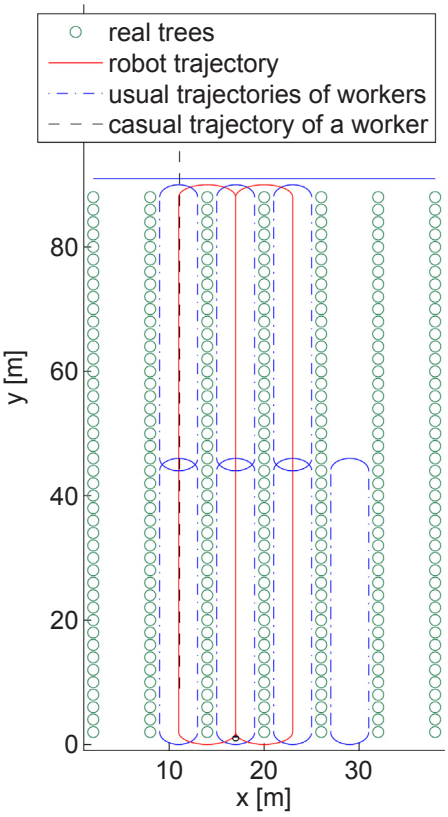


Fig. 7. Reference paths for the robot and the rural workers used for the second simulation. The trajectories are cyclical except the casual trajectory performed once by a worker.

$$\nu_c = \nu \frac{1}{1 + k_p \sqrt{p}} - k_{i,\nu} \|\mathbf{F_t}\|, \qquad \text{and} \qquad \omega_c = \omega - k_{i,\omega} \|\mathbf{F_r}\|, \tag{3}$$

where $\mathbf{u}_c = [\nu_c, \omega_c]^T$ is the control action sent to the vehicle, $k_p > 0$ is a design constant and $k_{i,\nu}$, $k_{i,\omega} > 0$ are the impedance constants.

Some problems related to the controllers' interaction can appear. For example, if an obstacle remains static while is detected by the laser, then an area with a high probability of collision is generated. Instead, if the obstacle is moving within the crop, a spatially distributed point cloud is generated, and therefore, the occupied area probabilities do not concentrate in the same place. A poor choice of the regulation factor can result in very deliberate evasions or low reduction of the velocity in high traffic areas. An adequate parameter selection can correct these problems.

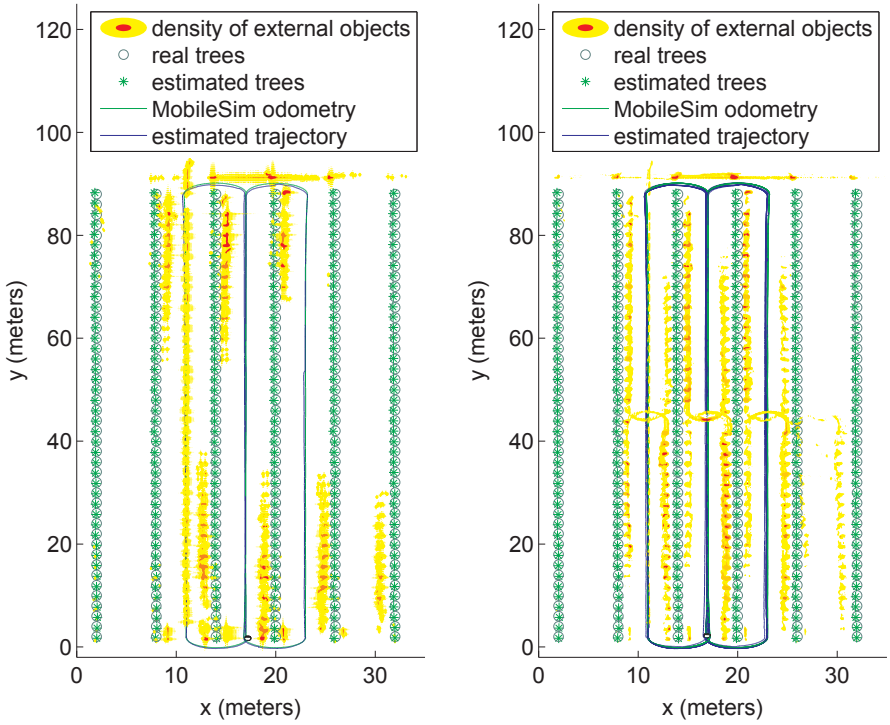
6. Simulations

Simulation results are presented to evaluate the proposed methodology. Matlab is used for implementing the mapping and control strategy, while the dynamic model of the vehicle is implemented in MobileSim Mobilesim. A robot Pioneer P3-AT Pioneer 3-at is used both in simulation as in experimentation. Besides, a C++ interface, based on shared memory, links the program running in the Matlab environment with the robot model of MobileSim. The interface also allows to read a virtual rangefinder and to generate the control command based on the reference velocities given by the controller.

Regarding the simulation environment, a virtual fruit grove with a regular geometry plus a Gaussian noise is generated. Moreover, an eight-shape trajectory along the grove trails is planned for the robot.

6.1. Weed mapping

In the first simulation, weeds are randomly placed around the trees in a simulated map. The design parameter is fixed to $\delta = 0.5$ m (defined



(a) Estimates obtained after performing only once the eight shape path.

(b) Estimates obtained after performing 10 times the eight shape path.

Fig. 8. KDE estimation of the worker presence density and SLAM estimations. Maps improve as data acquisition time increases, since iterative subsampling reduces outliers and replaces redundant data with worker observations in new areas.

Fig. 9. Experimental environment constructed using red traffic cones and stools as fruit trees and static obstacles. The robot must cyclically follow a path (blue line), evading a static obstacle in the first row, and a dynamic obstacle (worker going back and forth along the red line) in the second row. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

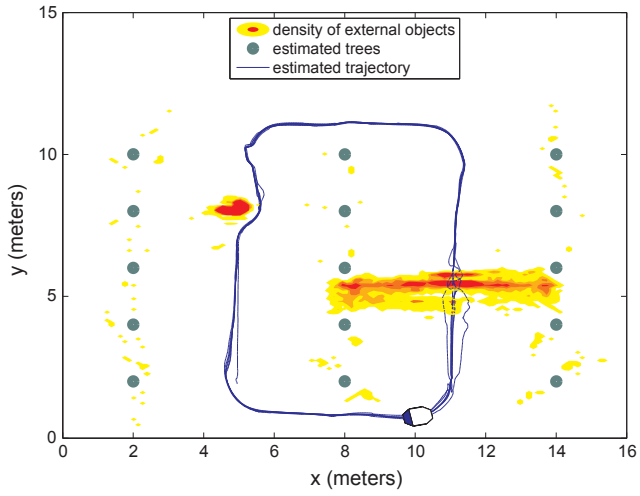


Fig. 10. Experimental results. The robot trajectory and trees positions estimated by SLAM are shown in blue and green, respectively. The probabilistic map of out-of-structure objects is displayed in color shades, where red areas highlight most likely zones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

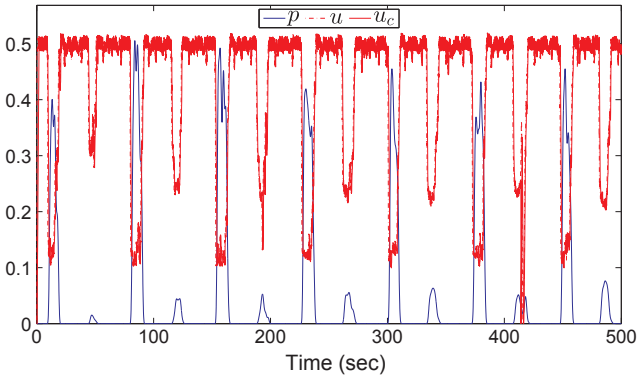


Fig. 11. Diagram of the linear velocity reduction according to the collision probability.

in Section 4), and no quota for the storage capacity is considered. Fig. 6(a) shows actual and estimated tree locations, the robot odometry obtained from MobileSim, the estimated trajectory and the KDE estimation of the weed density. In this case, the total size of the point cloud of the out-of-structure objects is 20,129 observations. On the other hand, the KDE estimation of the weed density when the storage capacity is bounded to 1000 observations is shown in Fig. 6(b). Note that the density estimations are similar, although only 4.96% of the data are used. This shows that deleted data were redundant since they did not provide significant additional information. Note also, how the subsampling reduces isolated observations. In addition, the map shown in

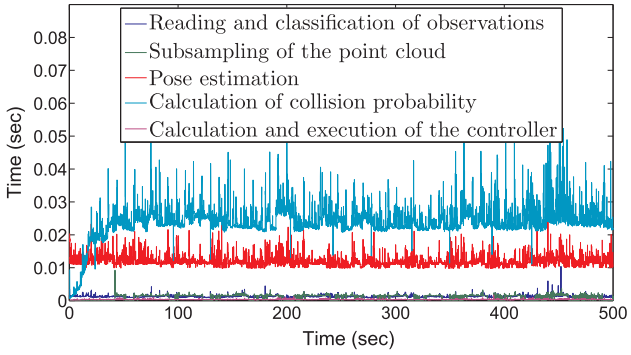


Fig. 12. Execution time of each algorithm. Most of the computer time is spent in probability calculation and SLAM, while the computer time required by subsampling is negligible. The computer time required by probability computes computation until the storage bound is reached, and from that moment on, the subsampling keeps it constant.

Fig. 6(b) covers about 3000 m^2 with 1000 data. If this space is divided into 1000 cells to build a grid-based map, each cell would have a resolution of approximately 3 m^2 which is not an adequate spatial resolution to the precise calculation of densities at specific locations. A zoom of Fig. 6(a) and (b) is showed in Fig. 6(c) and (d), respectively. Level curves of the estimated probabilistic map can be observed with more details in these figures. The highest concentration of weeds are plotted in red,[1] which can reveal a problem in the irrigation system.

6.2. Agricultural workers mapping

The second simulation is designed to show the potential of the proposed methodology to map the walking trails of the workers. In order to simulate rural workers walking along the fruit grove, nine robots are included in the MobileSim environment as shown Fig. 7. In this experiment, a virtual worker is programmed to follow a casual path at the beginning of the simulation, i.e., it only traverses its assigned path once in front of the robot and within the observation range of the laser. The remaining virtual workers follow their path throughout the whole simulation with stop periods of 50 s over each tree, as if they are performing pruning tasks. Gaussian noise is added to the desired path to avoid straight runs. The design parameter δ is fixed at 0.3 m, and a quota of 10000 observations is considered for the storage capacity.

Fig. 8 shows the real and estimated tree locations, the robot odometry obtained form MobileSim, and the estimated trajectory. The KDE estimations of workers presence density, after passing once and ten times over the crop following the eight-shaped path, are shown in Fig. 8(a) and (b), respectively. The more times the crop is traveled, the higher the accuracy of the estimated density map.

[1] For interpretation of color in Fig. 6, the reader is referred to the web version of this article.

In these plots, the concept of statistical significance is evidenced, since there are places with a high probability of finding a worker which are not adequately considered in Fig. 8(a) due to lack of significant data. Besides, Fig. 8(a) shows a high probability of workers presence in the areas where the virtual worker is observed following the casual path. The same areas in Fig. 8(b), after ten passes over the same path, present a low probability of workers' presence. This is because the virtual worker with a casual path is only observed during the initial pass and the subsampling process gradually eliminates these observations. The ability to forget atypical observations is another significant advantage of the methodology.

7. Experimental result

This Section presents the experimental results of the example application proposed in Section 5. In this case, a robot Pioneer 3-AT (P3-AT) from Omron Adept MobileRobots equipped with a laser range-finder sensor is used.

A real experimental environment requires days of experimentation for the robot to recognize areas with the highest traffic. Besides, it is necessary to distribute a large number of workers throughout the fruit grove so that navigation is not a mere path following. In order to reduce the experiment scale and to quickly show the probabilistic map effects in navigation, a scaled fruit grove is constructed using red traffic cones (as fruit trees) and stools (as obstacles) as shown Fig. 9. A path along the center line of the simulated crop is designed as the reference for the path following controller. A static obstacle is placed in the center of the first path and a dynamic obstacle (person pretending to be a rural worker) in the second path. Subsequently, the robot traveled around the diagrammed environment for 500 s avoiding obstacles and reducing its velocity in the areas with high traffic as is explained in Section 5.

The probabilistic map built by the robot can be observed in Fig. 10 where the robot trajectory estimation is shown in blue and the tree location estimations are shown in green. The probabilistic estimation of out-of-structure objects is displayed in color shades, where the red represents the more likely zones, the white represents the unlikely zones, and the other colors show zones with intermediate probability. The figure also shows how the robot avoids the obstacles by using the impedance-based controller.

A comparative graph showing the reference and actual linear velocities, and the collision probability calculated online while navigating are displayed in Fig. 11. In this figure, a strict relationship between these quantities can be observed. Note that static obstacles generate greater collision probabilities than dynamic obstacles because each static obstacle concentrates their observations on a site. For this reason, the velocity reduction produced by the collision probabilities must be designed in such a way that the robot reduces its velocity enough in traffic zones, and it does not stop in areas with static obstacles. The reduction factor used in this paper (see step 7 of Algorithm 2) decreases the velocity in zones of higher traffic and saturates asymptotically the reduction in zones with static obstacles. Besides, the linear velocity is set to zero when an obstacle is very close to avoid an imminent collision.

The runtime of each stage of the proposed methodology is shown in Fig. 12. Note the short runtime of the tasks: reading and classification of observations, subsampling of the point cloud, and calculation and execution of the controller. In particular, note that the bounded database and the recursive subsampling can be maintained and performed without computational difficulties. The runtime of the collision probability calculation increases at the beginning until the storage quota is reached, and then it remains constant. This allows regulating the quota according to the storage and processing capacities. Finally, note that the runtime of the robot pose estimation does not vary throughout the experimentation.

8. Conclusions

In this paper, a methodology for probabilistic mapping of out-of-structure objects in regular plantations such as fruit groves is presented. The aim of generating such a map is to include extra functionality for the robot to improve the Human-Robot Interaction. The proposed methodology could be incorporated as an extra functionality in the robot control system so that it performs its primary task in a better way. An example application is presented in which a robot, whose main task is to follow a trajectory within the fruit grove, reduces its velocity according to a collision probability calculated online. This is a cautious motion objective within an environment frequented by human workers. Similar behavior could be considered related to machinery working in the place.

The proposal discriminates the observations according to whether they come from trees or out-of-structure objects. Based on the tree observations, a SLAM is performed for localization and for correcting mapping errors with respect to the initial regular structure. The database, generated from the out-of-structure observations, is kept bounded, updated and representative thanks to a novel recursive subsampling method. This database allows an online probabilistic mapping of the environment based on a Kernel nonparametric estimator. The proposed methodology maps with a better spatial resolution and has theoretical convergence advantages in comparison with the grid-based estimators.

The simulations and the experimentation of the example application show the potentialities of the methodology.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compag.2018.05.018.

References

Andaluz, V.H., Roberti, F., Toibero, J.M., Carelli, R., Wagner, B., 2011. Adaptive Dynamic Path Following Control of an Unicycle-Like Mobile Robot. Springer, Berlin, Heidelberg.

Auat Cheein, F., Herrera, D., Gimenez, J., Carelli, R., Torres-Torriti, M., Rosell-Polo, J.R., Escolà, A., Arnó, J., 2015. Human-robot interaction in precision agriculture: sharing the workspace with service units. In: IEEE International Conference on Industrial Technology (ICIT), Sevilla, Spain, 2015.

Autor, D.H., 2015. Why are there still so many jobs? The history and future of workplace automation. J. Econ. Perspect. 29 (3), 3–30.

Bechar, A., Eben-Chaime, M., 2014. Hand-held computers to increase accuracy and productivity in agricultural work study. Int. J. Prod. Perform. Manage. 63 (2), 194–208.

Bechar, A., Vigneault, C., 2016. Agricultural robots for field operations: concepts and components. Biosyst. Eng. 149, 94–111.

Bechar, A., Yosef, S., Netanyahu, S., Edan, Y., 2007. Improvement of work methods in tomato greenhouses using simulation. Trans. Am. Soc. Agric. Biol. Eng. (ASABE) 50 (2), 331–338.

Bergerman, M., Maeta, S.M., Zhang, J., Freitas, G.M., Hamner, B., Singh, S., Kantor, G., 2015. Robot farmers: autonomous orchard vehicles help tree fruit production. IEEE Robot. Autom. Mag. 22 (1), 54–63.

Botev, Z.I., 2007. Kernel density estimation using Matlab. URL < http://www.mathworks.us/matlabcentral/fileexchange/authors/27236 > .

Botev, Z.I., Grotowski, J.F., Kroese, D.P., 2010. Kernel density estimation via diffusion. Ann. Stat. 38 (5), 2916–2957. http://dx.doi.org/10.1214/10-AOS799.

Ceres, R., Pons, F.L., Jimenez, A.R., Martin, F.M., Calderon, L., 1998. Design and implementation of an aided fruit-harvesting robot (Agribot). Ind. Robot 25 (5), 337–346.

Dammer, K.-H., 2016. Real-time variable-rate herbicide application for weed control in carrots. Weed Res. 56 (3), 237–246.

Dautenhahn, K., Billard, A., 1999. Bringing up robots or – the psychology of socially intelligent robots: from theory to implementation. In: Heidelberg, S.B. (Ed.). Proceedings of the Third International Conference on Autonomous Agents, Seattle, Washington, USA, pp. 366–367.

Edan, Y., Bechar, A., 1998. Multi-purpose agricultural robot. In: Sixth IASTED International Conference, Robotics and Manufacturing, Banff, Canada.

Ellis, C.A., Gibbs, S.J., Rein, G., 1991. Groupware: some issues and experiences. Commun. ACM 34 (1), 39–58.

Employment Projections Program, 2017. U.S. Bureau of Labor Statistics, Employment projections, employment by major industry sector (Oct. 2017). < https://www.bls.gov/emp/ep_table_201.htm > .

Gimenez, J., Herrera, D., Tosetti, S., Carelli, R., 2015. Optimization methodology to fruit grove mapping in precision agriculture. Comput. Electron. Agric. 116, 88–100. http://dx.doi.org/10.1016/j.compag.2015.06.013.
Györfi, L., Kohler, M., Krzyzak, A., Walk, H., 2002. A Distribution-Free Theory of Nonparametric Regression. Springer Series in Statistics Springer.
Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton. Robots 34 (3), 189–206.
Iida, M., Suguri, M., Uchida, R., Ishibashi, M., Kurita, H., Won-Jae, C., Masuda, R., Ohdoi, K., 2013. Advanced harvesting system by using a combine robot. Int. Fed. Autom. Control (IFAC) Proc. 46 (4), 40–44.
Kaldestad, K., Hovland, G., Anisi, D., 2012. 3d sensor-based obstacle detection comparing octrees and point clouds using CUDA. Model. Identif. Control 33 (4), 123–130.
Latif, H.O., Sherkat, N., Lotfi, A., 2009. Fusion of automation and teleoperation for person-following with mobile robots. In: International Conference on Information and Automation (ICIA), pp. 1240–1245.
Mitsunaga, N., Smith, C., Kanda, T., Ishiguro, H., Hagita, N., 2008. Adapting robot behavior for human-robot interaction. IEEE Trans. Robot. 24 (4), 911–916.
Mobilesim. URL < http://robots.mobilerobots.com/wiki/MobileSim > .
Panetta, F.D., 2015. Weed eradication feasibility: lessons of the 21st century. Weed Res. 55 (3), 226–238.
Paustian, M., Theuvsen, L., 2017. Adoption of precision agriculture technologies by German crop farmers. Precis. Agric. 18 (5), 701–716.
Perez-Ruiz, M., Slaughter, D.C., Fathallah, F.A., Gliever, C.J., Miller, B.J., 2014. Co-robotic intra-row weed control system. Biosyst. Eng. 126, 45–55.
Pierpaoli, E., Carli, G., Pignatti, E., Canavari, M., 2013. Drivers of precision agriculture technologies adoption: a literature review. Procedia Technol. 8, 61–69.
Pioneer 3-at. URL < http://www.mobilerobots.com/ResearchRobots/P3AT.aspx > .
Scursoni, J.A., Catanzaro, A.M., Catanzaro, M.P., Quiroga, J., Goldar, F., 2011. Evaluation of post-emergence herbicides for the control of wild oat (*Avena fatua* L.) in wheat and barley in Argentina. Crop Prot. 30, 18–23.
Secchi, R.C.H., Mut, V., 2003. An experience on stable control of mobile robots. Latin Am. Appl. Res. 33 (4), 379–385.
Tervo, K., Koivo, H.N., 2014. Adaptation of the human-machine interface to the human skill and dynamic characteristics. IFAC Proc. 47 (3), 3539–3544.
Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.
Torres-Sospedra, J., Nebot, P., 2014. Two-stage procedure based on smoothed ensembles of neural networks applied to weed detection in orange groves. Biosyst. Eng. 123, 40–55.
U. Nations, Sustainable development goals. 17 goals to transform our world. < http://www.un.org/sustainabledevelopment/development-agenda/ > .
Wand, M.P., Jones, M.C., 1993. Comparison of smoothing parameterizations in bivariate kernel density estimation. J. Am. Stat. Assoc. 88 (422), 520–528.
Yahya, N., 2018. Agricultural 4.0: Its Implementation Toward Future Sustainability. Springer, Singapore pp. 125–145.
Zimdahl, R., 2007. Fundamentals of Weed Science. Elsevier Science.