Computers and Electronics in Agriculture 78 (2011) 195–207

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection

F. Auat Cheein [a,*], G. Steiner [b], G. Perez Paina [b], R. Carelli [a]

[a] Instituto de Automatica, National University of San Juan, San Juan, Argentina
[b] Research Center in Informatics for Engineering, National Technological University, Cordoba, Argentina

ARTICLE INFO

Article history:
Received 12 March 2011
Received in revised form 1 June 2011
Accepted 22 July 2011

Keywords:
Agricultural mapping
SLAM
Mobile robot

ABSTRACT

Precision agricultural maps are required for agricultural machinery navigation, path planning and plantation supervision. In this work we present a Simultaneous Localization and Mapping (SLAM) algorithm solved by an Extended Information Filter (EIF) for agricultural environments (olive groves). The SLAM algorithm is implemented on an unmanned non-holonomic car-like mobile robot. The map of the environment is based on the detection of olive stems from the plantation. The olive stems are acquired by means of both: a range sensor laser and a monocular vision system. A support vector machine (SVM) is implemented on the vision system to detect olive stems on the images acquired from the environment. Also, the SLAM algorithm has an optimization criterion associated with it. This optimization criterion is based on the correction of the SLAM system state vector using only the most meaningful stems – from an estimation convergence perspective – extracted from the environment information without compromising the estimation consistency. The optimization criterion, its demonstration and experimental results within real agricultural environments showing the performance of our proposal are also included in this work.

1. Introduction

In this work, we present a Simultaneous Localization and Mapping (SLAM) algorithm based on stems detection for the creation of agricultural maps. An unmanned car-like mobile robot navigates the agricultural environment while it creates a map of such environment. That map is then stored for precision agricultural processes.

The SLAM algorithm is a recursive estimation process that simultaneously minimizes both: the vehicle's localization errors and the mapping errors of the environment which the mobile agent is navigating through. The SLAM algorithm was formerly introduced by Chatila and Laumond (1985) and Ayache and Faugeras (1989). In these works, the environment information was first included within the movements of the mobile robot while the automata was interacting with the environment at the same time. The works of Durrant-Whyte and Bailey (2006a,b) offer a concise introduction to the SLAM algorithm origins.

Up to these days, the SLAM problem can be efficiently solved by means of different filters available in the scientific literature. The most used method to solve the SLAM algorithm (Durrant-Whyte and Bailey, 2006a) is the Extended Kalman Filter (EKF). When an EKF-based SLAM is implemented, the system state vector is composed by both: the pose of the vehicle – its position and orientation within the environment – and the parameters that define the features from the environment. In Garulli et al. (2005), for example, an EKF-SLAM is implemented to model the environment exclusively using lines. Therefore, the mobile robot acquires only the geometric information from the environment that could be modeled by lines (such as walls, doors, etc.). The work of Guivant et al. (2000) shows the implementation of an EKF-SLAM based on point features extraction (such as trees, corners, etc.). Thus, the map created by a SLAM algorithm can be composed by any element from the environment that could be mathematically modeled. The main disadvantage of the EKF-SLAM is its processing time (whose complexity polynomially increases with the number of features (Durrant-Whyte and Bailey, 2006b)) associated with its correction stage.

As an alternative to the EKF to solve the SLAM problem, the UKF (Unscented Kalman filter, Thrun et al. (2005)) shows a better performance than the EKF when dealing with the nonlinearities associated with the process and observation models. On the other hand, the EIF (Extended Information Filter) improves the processing time of the correction stage of the SLAM algorithm, due to its linear computational cost (Thrun et al., 2005). Therefore, the EIF is more appropriate for real time applications.

Another solution to the SLAM algorithm is the Particle Filter (PF). Unlike the EKF, the PF is not restricted to Gaussian processes

* Corresponding author.
E-mail addresses: fauat@inaut.unsj.edu.ar (F. Auat Cheein), gsteiner@scdt.frc.utn.edu.ar (G. Steiner), gperez@scdt.frc.utn.edu.ar (G. Perez Paina), rcarelli@inaut.unsj.edu.ar (R. Carelli).

0168-1699/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2011.07.007

and it has a better managing of non-linearities involved in the estimation process (Thrun et al., 2005). Although, the real time implementation of the PF is still restricted. Despite of the nature of the filter used to solved the SLAM problem, the solution should ensure the consistency of the estimated map and the convergence of the estimation process (Dissanayake et al., 2001).

The nature of the features extracted from the environment to be used within the SLAM algorithm intrinsically depends on the type of sensor used. Thus, a vision sensor does not acquire the same information than an ultrasonic – or laser – range sensor or than a GPS (Global Positioning System), a gyroscope, an accelerometer, etc.

The use of vision systems allows the extraction of features from the environment – such as texture, color and shapes – that can be considered as supplementary information for the SLAM algorithm (Gonzalez and Wooks, 1993). The use of cameras and stereo vision systems is currently a research trend within the mobile robot SLAM field (Chekhlov et al., 2007; Flint et al., 2010).

Image classification techniques such as support vector machine (SVM, Haykin (1999)) allows the robust detection of an object of interest in the given scene. The SVM is a supervised learning method whose training stage-based on positive and negative samples of the object of interest-allows the learning of a general model of such object. Thus, the SVM is then able to classify new objects. The last is accomplished by the partition of a feature space where the objects of the scene are represented. This representation of the objects within the feature space is known as *descriptor*, where the Histogram of Oriented Gradients (HOG, He et al. (2008)) is one of the most used descriptors presented in the scientific literature.

In this work we present the implementation of an EIF-based SLAM for agricultural environments. The SLAM algorithm also has an optimization criterion associated with the information matrix of the EIF which allows the selection of the most significant features of the environment from a SLAM convergence perspective. Only these selected features will be used by the correction stage of the SLAM algorithm, although the mapping process remains unchanged for all features from the environment. The SLAM algorithm implemented in this work is oriented to precision agricultural environments (olive groves).

The features extracted from the environment correspond to olive stems. The olive stems are modeled as point-based features (Masson et al. (2005)). The stems extraction procedure is based on a vision sensor and a range laser sensor. The monocular vision system uses a Support Vector Machine for detection of stems from the images captured from the environment. From the detection of stems within an image, the implemented system estimates then the orientation of the stem with respect to the mobile robot (angle information). The range laser sensor returns the distance information from the mobile robot to the detected stem after searching in a neighborhood of the angle information provided by the SVM. The range and angle information is then sent to the SLAM algorithm which uses them to estimate the position of the mobile robot within the environment and to construct a map of such environment. Real experimental results performed at the National Institute of Agricultural Technology (INTA), Argentina, are presented to show the advantages of our proposal.

This work is organized as follows. Section 2 shows the recent works published in the field related to our proposal; Section 3 shows the general system architecture implemented in this work; Section 4 shows the kinematic model of the mobile robot used; Section 5 presents the stems extraction procedures based on both: artificial vision and laser; Section 6 shows the SLAM structure and Section 7 shows the EIF-based SLAM. Section 8 shows the experimental results of our work and Section 9 presents the conclusion and discussion of this article.

2. Related work

The mobile robot applications for precision agriculture is an increasing field within the robotic community. For example, the work of Cariou et al. (2009) presents a navigation strategy for a four-wheeled car-like mobile robot, which is kinematically compatible with the mobile robot's kinematic constrains. This strategy allows the automata to maneuver in order to accomplish accurate tasks. Also, the work of Benton Derrick and Bevly (2009) presents a solution to the navigation of a farm tractor based on an adaptive control strategy. The main difference with Cariou et al. (2009) is that the work of Benton Derrick and Bevly (2009) solves the navigation problem using control laws whereas Cariou et al. (2009) uses path planning techniques. The work of Johnson et al. (2009) presents a multi-agent system of mobile robots for harvesting. These works do not have into account the current map information in order to plan the mobile robot actions.

The work of Nagasaka et al. (2009) presents an autonomous mobile robot whose navigation is based on GPS and inertial measurements. Therefore, no map is managed during the mobile robot task. On the other hand, the work of Jin and Tang (2009) introduces a mapping strategy based on detection of corn plants using a stereo camera. Although Jin and Tang (2009) efficiently extracts the features from the environment (corn plants), the localization of the mobile robot is not considered during the mapping procedure. Also, the lightening conditions of the environment compromise the certainty of the features detection.

The SLAM algorithm is still an open issue in precision agriculture. Most of the SLAM developments of the last years have been focused in solving the outdoor mapping for unstructured environments. In this line of research, the work of Nüchter et al. (2007) introduces an SLAM algorithm with six degrees of freedom to map 3D environments. This algorithm is appropriate for aerial SLAM and therefore for extensive plantation reconstruction, although it is unable to characterize features from such plantation. On the other hand, the work of Schleicher et al. (2009) presents an urban SLAM which localization method is based on GPS measurements and the features extraction procedure consist of patterns extracted from an stereo vision system associated with natural landmarks from the environment. Also, the work of Cole and Newman (2006) presents an outdoor reconstruction based on 3D laser data. In this work, the processing time becomes crucial for large environments and features extracted are no associated with plantation type features.

Our work studies both: the localization and the mapping problem of a mobile robot navigating within an olive grove while detecting the olive stems of the environment. This way, a map of the environment can be automatically produced and updated during the navigation of the robot. The localization and mapping problem is solved by the implementation of an optimized EIF-SLAM algorithm. The olive stems detection is based on a SVM implemented on a monocular vision system and a range laser sensor.

3. General system architecture

The general architecture of the system proposed herein is presented in Fig. 1. It can be briefly described as follows.

(i) The agricultural environment is an olive grove placed at the INTA (*National Institute of Agricultural Technology*), Argentina.
(ii) The *vision* and *laser* blocks refer to the devices used for the detection of olive stems from the environment. The monocular vision system estimates the orientation of the stem position from the camera point of view. The range sensor laser provides the range information associated with each

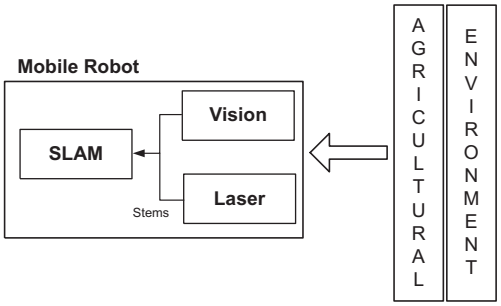

Fig. 1. General system architecture.

detected stem from the monocular vision system. The range and orientation information are considered as parameters associated with the feature *stem*. These parameters are then passed to the SLAM algorithm.

(iii) The SLAM algorithm recursively estimates the map of the environment and the pose of the vehicle within that environment.

In the following sections, each block of Fig. 1 will be explained in detail.

4. Mobile robot model

The vehicle used in this work is an unmanned non-holonomic car-like type mobile robot. Eq. (1) shows the kinematic equation associated with the robot; Fig. 2 shows the graphical representation of the robot, where r is the minimum radius of turning of the mobile robot and L is the distance between axes; V is linear velocity of the vehicle and θ is the vehicle's heading; ψ is the orientation of the mobile robot within the environment. The motion control command of the robot is defined by its linear velocity and its heading ($u = [V\theta]^T$).

$$\begin{cases} \dot{x}_{robot} = V\cos(\psi) \\ \dot{y}_{robot} = V\sin(\psi) \\ \dot{\psi} = \frac{V}{L}\tan(\theta) \end{cases} \tag{1}$$

5. Features extraction

In this work, the features extracted from the environment correspond to the stems of the olive grove. The stems are defined by two parameters: range and angle (Masson et al., 2005). The range is the distance from the stem to the mobile robot position whereas the angle is the orientation of the stem position with respect to the pose of the vehicle. Fig. 3 shows the graphical representation of the stem's parameters. In addition, Eq. (2) shows the analytical model of the stems which are considered as point-based features by the SLAM algorithm – see Masson et al. (2005).

$$z_{stem} = h_i[x_{robot}, y_{robot}, \psi, x_{stem}, y_{stem}, w] = \begin{bmatrix} z_R \\ z_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{robot} - x_{stem})^2 + (y_{robot} - y_{stem})^2} \\ \arctan\left(\frac{y_{robot} - y_{stem}}{x_{robot} - x_{stem}}\right) - \psi \end{bmatrix} + \begin{bmatrix} w_R \\ w_\beta \end{bmatrix} \tag{2}$$

In Eq. (2), z_{stem} is the vector with the parameters associated with the stem of an olive tree. As was stated before, z_{stem} is composed by the range information (z_R) and the angle information (z_β). x_{robot}, y_{robot} and ψ represent the position and orientation of the robot within the environment whereas x_{stem} and y_{stem} represent the Cartesian coordinates of a stem with respect to a local reference system attached to the mobile robot. w is the Gaussian vector noise associated with the measurement procedure. Fig. 3 shows a graphical representation of the variables involved in Eq. (2). As it can be seen, the stem is considered as a point-based feature (Masson et al., 2005).

The features extraction procedure is divided into two phases: the angle extraction of each stem and the range extraction of each stem. The angle extraction of a stem is performed by the processing of images acquired by the monocular vision system used in this work. A support vector machine (SVM) is implemented in this work to detect the stems of the olive trees. The SVM stem detection method gives the position of the stems within the image. From the stem position and using projective geometry, it is calculated the angle of the stem position with respect to the principal axis of the camera. Due to the fact that in this way we can only calculate angle measurements, a range laser sensor is used to estimate the distance of the detected stem from the mobile robot. The use of two sensor devices for range and angle measurements of a stem is based on the following:

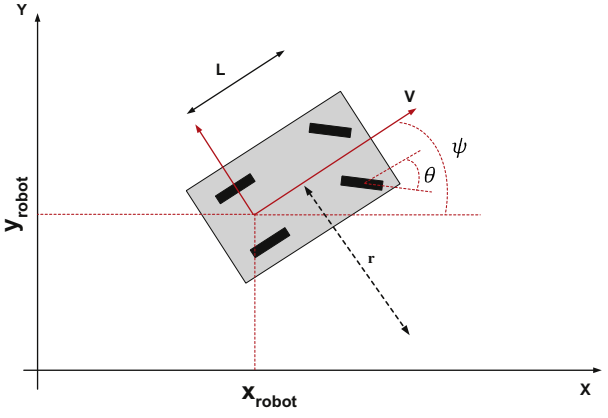

Fig. 2. Graphic representation of a car-like robot used in this work.

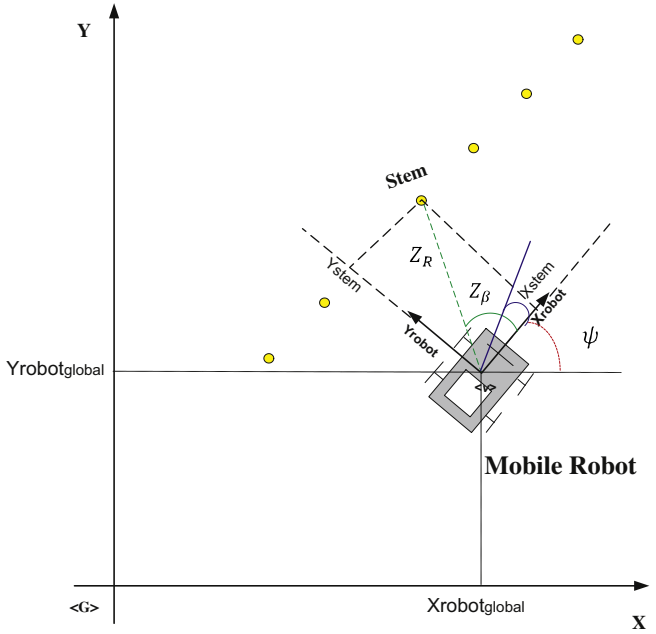

Fig. 3. Graphic representation of the stems in the olive grove environment.

(i) A single monocular vision camera does not return depth information.
(ii) A range laser sensor is able to acquire range and angle measurements although it cannot distinguish between tree stems, people or any other obstacle with a shape morphology equivalent to a stem.

Following, each of the features extraction method used in this work is explained.

5.1. Monocular vision-based stem detection

For the monocular vision-based olive stem detection we have implemented machine learning techniques for environment recognition. Environment recognition techniques are usually used to build a map of structures, rigid objects, people, etc., surrounding the mobile robot. The environment recognition task can be done with most kind of exteroceptive sensors, such as cameras, ultrasonic or laser range finder sensors. Particularly, in the present work a monocular camera is olive stem computer vision-based detection.

The environment recognition process can be divided into two stages:

(i) Features extraction.
(ii) Classification using the extracted features.

Feature space representation is performed by means of HOG (*Histograms of Oriented Gradients*, Dalal and Triggs (2005)). This method is commonly used for people detection and it is based on a simplified version of SIFT descriptor (Lowe, 2004), applied in the image. In order to classify objects of interest within the image, in this work we use the SVM learning technique.

5.1.1. HOG descriptors

HOG descriptors (Dalal and Triggs, 2005) are used to represent the image into a feature space of lower dimensionality. The following steps are carried out to obtain a HOG descriptor.

(i) Compute the spatial derivative of the image in both x-axis and y-axis direction, which are used to get the gradient magnitude and direction at each image point.
(ii) Divide the image into block descriptors of fixed size. These blocks are distributed to cover the whole image overlapped in a fixed number of pixels.
(iii) Each block is divided into cells.
(iv) A vector $\mathbf{v}$ is computed for each cell whose elements represent gradient directions (bins). Each gradient pixel of the cell contributes to elements of this vector according to its magnitude and direction.
(v) Due to a large variations in the lighting and contrast conditions it is necessary to normalize this vector. Using the L_2-*norm*, proposed in Dalal and Triggs (2005) to normalize this vector such that L_2-*norm* is $\|\mathbf{v}\|_2$ and adding and small constant ε to avoid division by zero, the normalized vector is

$$\mathbf{v} \leftarrow \frac{\mathbf{v}}{\sqrt{\|\mathbf{v}\|_2^2 + \varepsilon^2}}$$

(vi) The vectors of bins representing each cell in a block are concatenated, resulting in a vector block descriptor.
(vii) Finally, these vector blocks are grouped in a single image descriptor vector.

Fig. 4 shows the process of the HOG transformation described above.

5.1.2. SVM classification

Support Vector Machines (SVM, Vapnik (1998)) are kernel machines (Müller et al., 2001) used in supervised learning for classification tasks. A classification task usually consists of two stages: the training stage and the test or evaluation stage. The training stage data consists of a set of labeled target objects and a set of attributes or features. The goal of the SVM is to generate a model with the ability to classify objects of the testing data using only features that represent these objects.

Formally, given a set of n training data $\mathcal{D}$ of the form

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, \quad y_i \in \{-1, 1\}\}_{i=1}^n$$

where y_i are the class of given $\mathbf{x}_i$, and $\mathbf{x}_i$ are real vectors of dimension p, the SVM gives the hyperplane that maximizes the margin between these two classes ($y_i = \pm 1$).

Given a set of separable points, there exists an hiperplane π:$\mathbf{w} \cdot \mathbf{x} + b = 0$ which divides these vectors $\mathbf{x}_i$, $i = 1, \ldots, n$ into two different groups. Therefore, the SVM solves the following quadratic optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.a.} \quad & y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geqslant 1 \end{aligned} \tag{3}$$

Non linear classification can be solved by changing the dot product by a kernel function. The kernel is related to the transformation $\phi(\mathbf{x}_i)$ by $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.

5.1.3. Implementation issues

In order to obtain the best image descriptor, the HOG implementation uses a set of parameters that needs to be tuned. In the present work the following parameters were used,

- block size (2×2 cells),
- block overlap (8 pixels),
- cell size (8×8 pixels),
- numbers of bins (9 bins, with only the direction of the gradient, which gives 9 bins in 180 degrees),
- image size to be describe (32×72 pixels).

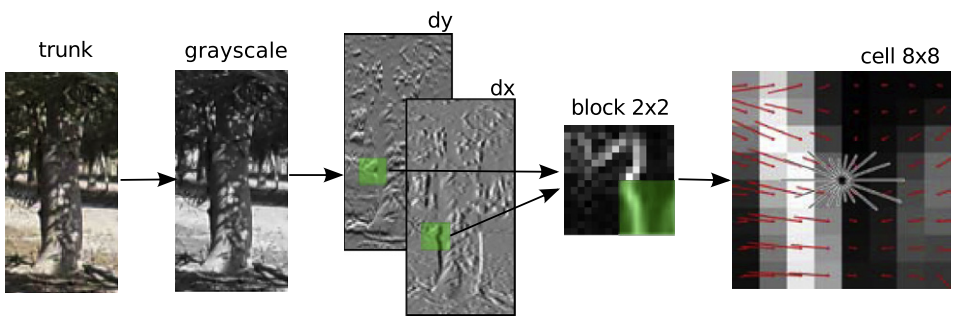

Fig. 4. Details of the 8×8 cell with the vector descriptor of 8 bins in a 2×2 block.

The SVM classification uses a linear kernel, which was previously trained with a positive images set (with olive stems) and a negative set (without olive stems). Training images and the subsequent classification should be of the same size. For any other image size a grid is generated that partitions the image in blocks of the appropriate size. These blocks are classified separately.

Another issue associated with the implementation is the scale. Searching in a grid of a fixed size, only the olive stems of the training image size can be detected. To circumvent this, multi-scale descriptions were used, i.e., we have used a pyramid of images (Lindeberg, 1994) with a series of smoothing filters and downsampling, where each level of this pyramid represents a scale of the original image. Applying the searching criterion of the image grid to each level, gives a set of olive stems detections at different scales. Fig. 5 shows an example of the SVM classification of a picture taken at the olive grove from the *INTA*. As it can be seen, several stems are effectively detected.

5.1.4. Angle measurement

The bearing or angle measurement is based on the pin-hole model of a perspective camera. The scene projection in the image plane is represented by a 3×4 homogeneous matrix known as camera projection matrix – see Eq. (4).

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}, \quad -\mathbf{C}] \tag{4}$$

where $\mathbf{K}$ is the intrinsic parameters matrix; $\mathbf{R}$ (the rotation matrix) and $\mathbf{C}$ (optical center) are the extrinsic parameters. The projection matrix has 11 degrees of freedom. In general, the camera projection matrix is a 3×4 full-rank matrix and, being homogeneous, it has 11 degrees of freedom. The camera parameters can be obtained by a calibration process (Zhang, 1999).

The intrinsic parameter matrix is of the form:

$$\mathbf{K} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where the main parameters are the focal length f and the coordinate of the principal point (u_0, v_0).

Fig. 6 shows how vertical segments are projected onto the camera image plane. The 3D scene points $\mathbf{X}_i$ are projected to the image points $\mathbf{m}_i$ over the image plane Π, which is at a distance f of the camera coordinates system, so that the image plane is normal to the principal ray or principal axis. 3D points are projected to image points as

$$\mathbf{m}_i \simeq \mathbf{PX}_i = \mathbf{KR}[\mathbf{I}, -\mathbf{C}]\mathbf{X}_i = \mathbf{Kx}_i$$

where $\simeq$ means that they are equal but a given scale factor.

Given a line representing the object of interest in the image plane, and knowing the camera intrinsic parameters matrix, a point in this line can be transformed to the camera coordinates system by means of the following transformation: $\mathbf{x}_i \simeq \mathbf{K}^{-1}\mathbf{m}_i$,

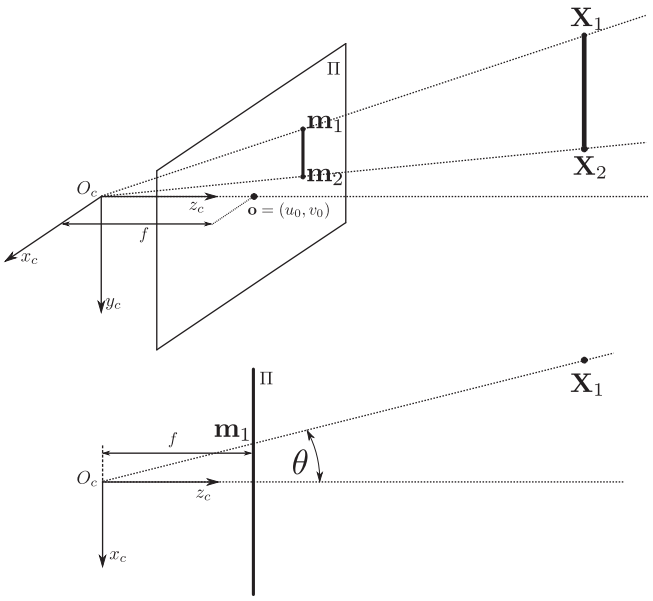

Fig. 6. Bearing measurement.

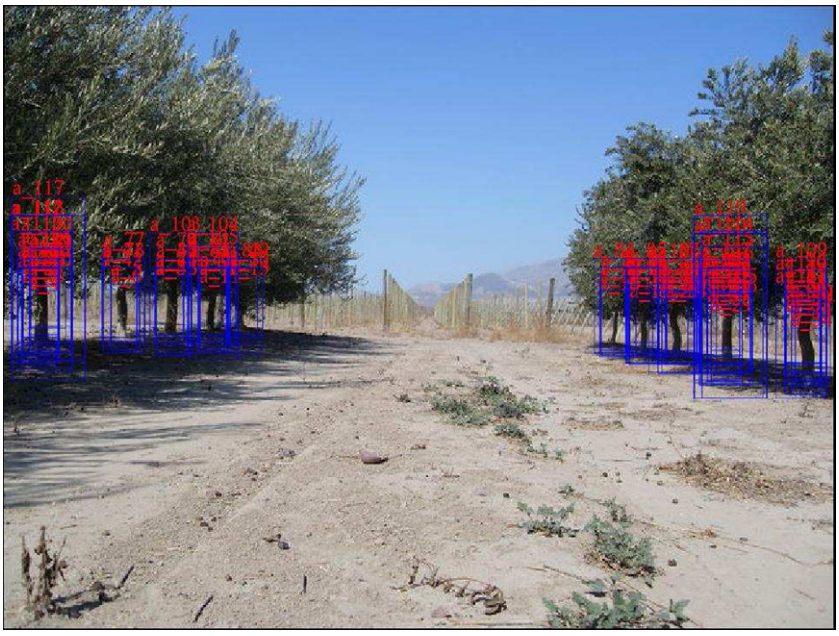

Fig. 5. SVM stems classification example. The successive blue squares circumscribing the stem represent the multi-scale stem detection. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $\mathbf{x}_i = [x_i\ y_i\ f]^T$. The angle or bearing of the vertical line can be compute as $\theta_i = tan^{-1}(x_i/f)$. This angle corresponds to the rotation around the y_c axis of the camera. Thus, considering that that line passes through the center of mass of the a detected stem from the environment, we can obtain the bearing of the stem with respect to the mobile robot's pose by using the procedure presented in this section.

5.1.5. Laser-based stem detection

The information extracted by the range laser sensor from the environment consists of the range information associated with the detection of olive stems. The range sensor used in this work is a range sensor laser built by **SICK**. This laser acquires 181 measurements between 0 and 180 degrees with a maximum range of 30 m. The stem extraction algorithm based on laser measurements was previously published by Masson et al. (2005) and Auat Cheein et al. (2010). The stem extraction algorithm consists of considering the stem as a point-based feature, as was stated in Eq. (2). This algorithm extracts the distance from the mobile robot to the stem using the range histogram associated with the laser measurements. This situation is shown in Fig. 7. Further information regarding this algorithm can be found in Masson et al. (2005) and Auat Cheein et al. (2010).

In order to integrate the laser-based stem detection algorithm with the monocular vision-based stem detection, we have used the following procedure.

(i) The system first detects the olive stems from an image acquired by the monocular vision sensor using SVM as was stated in 5.1 and returns the angle measurements associated with the center of mass of each detected stem.
(ii) Considering that the camera remains fixed to the vehicle, the angle associated with each detected stem from an image is directly associated with the local reference frame attached to the mobile robot (e.g., if a stem is detected to be 30 degrees with respect to the center of the image, it is also 30 degrees with respect to the robot's pose – see Section 8).
(iii) Once the angle of each detected stem is determined by the monocular vision features extraction method, the laser is used to confirm the presence of a stem in the neighborhood of the detected angle. Thus, if the monocular vision system finds a stem at 30 degrees, then the laser searches in a neighborhood of 30 degrees to find a possible stem using the method presented above. If a stem is found, then it is associated with the angle given by the monocular vision features extraction method and the range given by the range laser sensor method. If a single stem is not detected by both the vision and laser measurements, then it will not be used by the EIF-SLAM algorithm.
(iv) The process is repeated for all detected stems from the image.

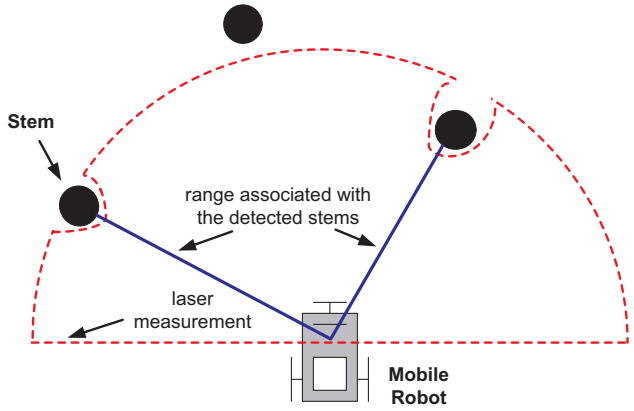

Fig. 7. Representative graphic of the laser-based stem detection used in this work.

6. SLAM algorithm

The SLAM algorithm implemented in this work is based on an Extended Information Filter (EIF). The EIF algorithm is shown in Eq. (6).

$$\begin{aligned}
\mu_{t-1} &= \Omega_{t-1}^{-1}\xi_{t-1} \\
\bar{\Omega}_t &= (G_t\Omega_{t-1}^{-1}G_t^T + Q_t)^{-1} \\
\bar{\xi}_t &= \bar{\Omega}_t g(u_t, \mu_{t-1}) \\
\bar{\mu}_t &= g(u_t, \mu_{t-1}) \\
\Omega_t &= \bar{\Omega}_t + H_t^T R_t^{-1} H_t \\
\xi_t &= \bar{\xi}_t + H_t^T R_t^{-1}[z_t - h(\bar{\mu}_t) + H_t\bar{\mu}_t]
\end{aligned} \tag{6}$$

In Eq. (6), μ is the system state vector which contains the information related to the pose of the vehicle within the environment as well as all the features extracted from such environment; ξ and Ω – the information matrix– are the parameters of the EIF, where ξ has the same dimension of μ and Ω has the same dimension of the covariance matrix associated with the system state vector; G is the Jacobian matrix associated with the model of the system process whereas Q is its covariance matrix; u represents the command control vector. H is the Jacobian matrix associated with the model of the measurements and R its covariance matrix; the current feature processed by the EIF-SLAM is denoted by z. Finally, $\bar{\xi}$ and $\bar{\Omega}$ are the predicted parameters of the EIF (further information regarding the EIF can be found in Thrun et al. (2005)).

The correction stage of the sequential EIF (Thrun et al., 2005) can be expressed as shown in Fig. 8. The correction stage of the SLAM algorithm is only performed with the features extracted from the environment that have a correct association with respect to the predicted features (Thrun et al., 2005; Durrant-Whyte and Bailey, 2006a). In this work, an optimization criterion is proposed within the correction stage of the sequential EIF-SLAM. The optimization criterion will be presented in Section 7.1.

Let N_t be the set of detected features at time instant t
Let $M_t \subseteq N_t$ be the set of features with correct association at time instant t
for $j = 1$ to $\#M_t$ **do**
  $\Omega_t = \bar{\Omega}_t + H_t^T R_t^{-1} H_t$
  $\xi_t = \bar{\xi}_t + H_t^T R_t^{-1}[z_j - h(\bar{\mu}_t) + H_t\bar{\mu}_t]$
  $\bar{\Omega}_t := \Omega_t;\ \bar{\xi}_t = \xi_t$
end for

Fig. 8. Algorithm for the correction stage of the sequential EIF-SLAM.

7. EIF derived from the Extended Kalman Filter

The parametrization of the EIF can be derived directly from the EKF formulation (Thrun et al., 2005). Thus,

$$\begin{cases} P_t = \Omega_t^{-1} \\ \xi_t = \Omega_t \mu_t \end{cases}. \tag{7}$$

In Eq. (7), P is the covariance matrix associated with the EKF system state vector (ξ). According to Thrun et al. (2005), the correction stage of the EIF is linear whereas the correction stage of the EKF is square (O^2) – thus, the EIF is *faster* than the EKF. This situation can be seen from Eqs. (6) and (8), where Eq. (8) represents the Extended Kalman Filter.

On the other hand, the prediction stage of the EIF requires the inversion of an $n \times n$ square matrix – where n is the dimension of the system state vector, whereas the prediction stage of the EKF is linear.

$$\begin{aligned} \hat{\xi}_t^- &= f(\hat{\xi}_t, u_t) \\ P_t^- &= A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \\ K_t &= P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \\ \hat{\xi}_t &= \hat{\xi}_t^- + K_t(z_t - h(\hat{\xi}_t^-)) \\ P_t &= (I - K_t H_t) P_t^- . \end{aligned} \tag{8}$$

In Eq. (8), f is the model of the process, $\hat{\xi}$ is the system state vector, u is the control commands vector, P is the covariance matrix associated with $\hat{\xi}$. A is the Jacobian matrix of f with respect to $\hat{\xi}$; Q represents the covariance matrix of the Gaussian noise of the process and W is its Jacobian matrix; K is the Kalman gain, H is the Jacobian matrix associated with the observation model and R is its associated covariance matrix. Vector z is the current feature of the environment extracted by the algorithm whereas h is its prediction. Further information of the EKF and its implementation in an SLAM algorithm can be found in Auat Cheein et al. (2010), Durrant-Whyte and Bailey (2006a) and Thrun et al. (2005).

7.1. EIF-SLAM optimization criterion

During the navigation of a mobile robot while performing the SLAM algorithm, not all the information acquired from the environment is useful for the estimation process, although all the information might be used during the mapping procedure. For example, if the system detects two olive stems, perhaps only one of them will contribute the most to the estimation process instead of the other. Thus, both stems should be used for the map construction of the plantation but only one should be used during the estimation process. This features selection criterion also contributes to decrease the computational costs of the correction stage of the SLAM because only the most meaningful features with correct association will be used instead of all detected features. In this work, we propose an optimization criterion based on a features selection procedure in order to decrease the SLAM computational costs without compromising the estimation convergence, as it will be shown herein.

According to Thrun et al. (2005) and Auat Cheein et al. (2010), when an EKF-SLAM is executed, the determinant of the covariance matrix of the SLAM system state converges to zero as time tends to infinity and the convergence of the estimation process is not lost, as it is shown in Eq. (9).

$$\lim_{t\to\infty} |P_t| = 0. \tag{9}$$

Changing P_t in Eq. (9) with the parameters of the EIF – see Eq. (7), we have that,

$$\begin{aligned} \lim_{t\to\infty} |P_t| &= \lim_{t\to\infty} \tfrac{1}{|\Omega_t|} = 0 \\ \lim_{t\to\infty} |\Omega_t| &= \infty \end{aligned} \tag{10}$$

Eq. (10) establishes that the determinant of the EIF information matrix tends to infinity as time also tends to infinity. Considering that P is symmetric and positive definite, then Ω – which is the inverse of P – is also symmetric and positive definite. According to the correction stage of the information matrix shown in Eq. (6), we have that Ω_t and $\bar{\Omega}_t$ are symmetric positive definite matrices; also, $H_t^T R_t^{-1} H_t$ is symmetric positive semi-definite matrix – due to the fact that R_t is a symmetric positive semi-definite matrix. Then, the following holds – see Harville (1997).

$$\begin{aligned} |\Omega_t| &= |\bar{\Omega}_t + H_t^T R_t^{-1} H_t| \\ |\Omega_t| &\geqslant |\bar{\Omega}_t| \\ |\Omega_t| &\geqslant |H_t^T R_t^{-1} H_t| \end{aligned} \tag{11}$$

Eq. (11) is consistent with the results shown in Auat Cheein et al. (2010) due to the fact that $|P_t| \leqslant |P_t^-|$ and replacing by Eq. (7) we have that: $|P_t| = \frac{1}{|\Omega_t|} \leqslant \frac{1}{|\bar{\Omega}_t|} = |P_t^-|$, which turns into: $|\Omega_t| \geqslant |\bar{\Omega}_t|$.

Let us consider the following theorem extracted from Harville (1997).

Theorem 1. Let A and B be two $n \times n$-positive definite matrices and let C be a $n \times n$-positive semi-definite matrix such that $A = B + C$. Then,

$$|A| \geqslant |B| \wedge |A| \geqslant |C|$$

Furthermore, let $\lambda_{A_i} > 0$, $i = 1 \ldots n$, be the set of eigenvalues of A – $eig(A)$; also, let $\lambda_{B_i} > 0$, $i = 1 \ldots n$, be the set of eigenvalues of B – $eig(B)$ – and $\lambda_{C_i} \geqslant 0$, $i = 1 \ldots n$ be the set of eigenvalues of C – $eig(C)$. Then, $|A| = \lambda_{A_1} \times \ldots \times \lambda_{A_n}$; $|B| = \lambda_{B1} \times \ldots \times \lambda_{B_n}$ and $|C| = \lambda_{C_1} \times \ldots \times \lambda_{C_n}$. By inspection, it is possible to see that if one of the eigenvalues of B increases its value, then $-B-$ also increases. Considering *Theorem I* the last implies that the lower bound of $-A-$ will also increase. Equivalent reasoning can be applied to matrix C.

According to the last paragraph and having into account Eq. (11), we can see that it is possible to increase the lower bound of $|\Omega_t|$ by increasing either $|\bar{\Omega}_t|$ or $|H_t^T R_t^{-1} H_t|$. However, $|\bar{\Omega}_t|$ remains constant during the correction stage for every observation z_j as it was shown in the algorithm of Fig. 8. Thus, only $|H_t^T R_t^{-1} H_t|$ is able to change with a given observation z_j.

Therefore, it is possible to establish an optimality criterion of the EIF-SLAM based on the selection of the most significant features. In this work, the *most significant features* will be those features that increase the most the lower bound of the determinant of the information matrix $|\Omega_t|$. In this way, the set M_t of features with correct association in the algorithm of Fig. 8 will be restricted to the most significant features. Eq. (12) shows the optimization criterion based on the selection of the most significant features.

$$z_i^{opt} : arg_z max(|\Omega_t|) = arg_z max(|H_t^T R_t^{-1} H_t|) \tag{12}$$

The EIF-SLAM correction stage algorithm with the optimization criterion proposed herein is shown in Fig. 9. The prediction stage of the EIF-SLAM remains unchanged.

In line of code 3 in Fig. 9, *LIM* is the maximum number of features with correct association that will be used by the correction stage – i.e., *LIM* = 2 means that the EIF-SLAM will correct only with the two most significant features (if they exist) from the convergence perspective of the estimation process; line of code 5 shows the determination of z_j^{opt} based on the optimization criterion shown in Eq. (12). In line of code 9, if z_j^{opt} is used in one iteration

Let N_t be the set of detected features at time instant t
Let $M_t \subseteq N_t$ be the set of features with correct association at time instant t
Let LIM be the maximum number of features to be used in the correction stage
for $j = 1$ to $min\{LIM, \#M_t\}$ **do**
  find $z_j^{opt} : arg_z max(|\Omega_t|)$
  $\Omega_t = \bar{\Omega}_t + H_t^T R_t^{-1} H_t$
  $\xi_t = \bar{\xi}_t + H_t^T R_t^{-1}[z_i^{opt} - h(\bar{\mu}_t) + H_t \bar{\mu}_t]$
  $\bar{\Omega}_t := \Omega_t$; $\bar{\xi}_t = \xi_t$
  $M_t = M_t - \{z_j^{opt}\}$
end for

Fig. 9. Algorithm of the optimized correction stage of the sequential EIF-SLAM.

of the *for*-loop, that feature is deleted from M_t and it will be not used again during the *for*-loop.

8. Material and methods

The experimental results were carried out at the National Institute of Technological Agriculture (*Instituto Nacional de Tecnología Agropecuaria*, INTA), Argentina. Fig. 10(a) shows a geo-referenced satellite image of the olive grove environment. The vehicle used is an unmanned car-like mobile robot, shown in Fig. 10(b). The robot is equipped with two laser range sensors mounted at the front part (built by SICK, see Fig. 10(c)). Only the laser pointing to the front of the vehicle was used in this work. The robot also has a *Bumblebee 2* stereo camera (see Fig. 10(d)) although only one camera was used in this work (because the camera aperture was appropriate for the experiments shown herein). The vehicle has two computers mini ITX dual atom processor on it, in which all the processing is performed on line.

As it can be seen in Fig. 10(b), the laser and the camera are align to a same vertical axis. Therefore, no further transformations are needed to manage the laser nor the vision information.

In addition, the maximum range of the laser was set to 30 m. Also, the maximum number of features to be used within the correction stage of the EIF-SLAM was set to three. That is, LIM = 3 in the algorithm of Fig. 9. Therefore, in this work we have used only the three most significant features – according to the optimality criterion proposed in Section 7.1 – to correct the EIF-SLAM estimation process. The entire sampling time of the system was of 1 s. This sampling time involves the features extraction procedures and the SLAM prediction-correction stages. The maximum linear speed of the robot was set to 0.2 m/s and the maximum angular velocity was of 10 deg/s. Also, the EIF-SLAM algorithm was executed in real-time on the mobile robot.

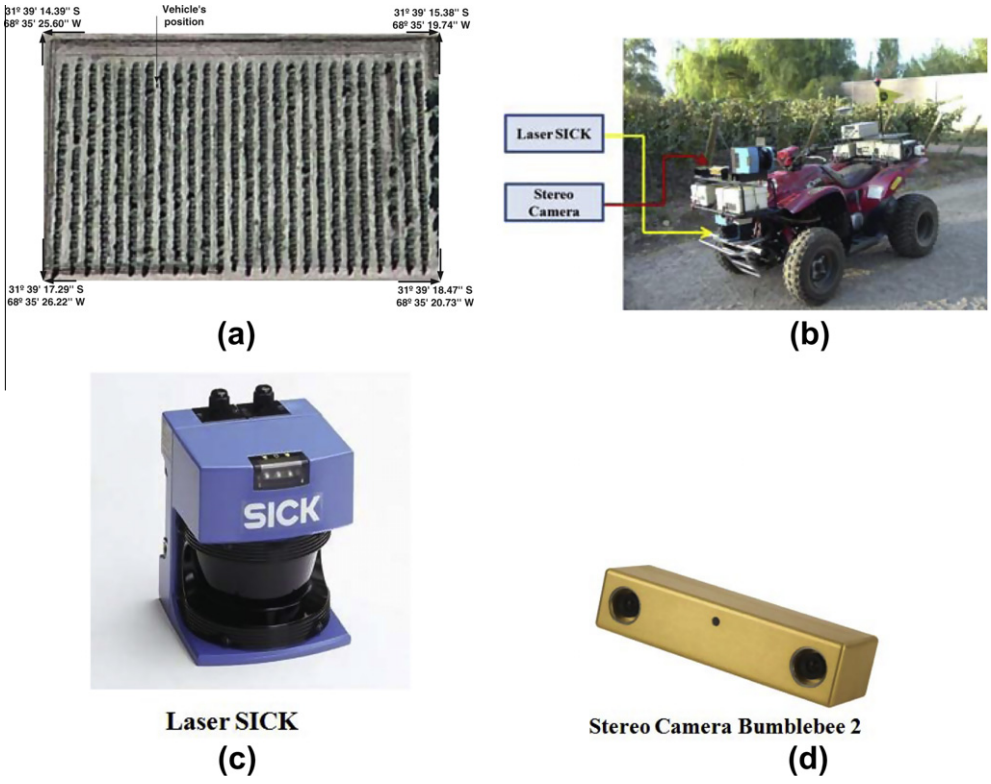

Fig. 10. Materials for the experimentation stage. (a) A geo-referenced image of the INTA's olive grove where the experimentation was carried out; (b) the unmanned car-like mobile robot used in this work; (c and d) the range laser sensor and the camera sensor used herein.

8.1. EIF-SLAM results

Fig. 11 shows four cases of visual-based stem detection. The four cases correspond to the olive grove shown in Fig. 10(a).

On the other hand, Fig. 12 shows two environment reconstructions based on the odometric information of the mobile vehicle. As it can be seen, the odometric information is noisy and cannot be used for localization purposes. The initial pose of the robot was set to $[x_{robot}, y_{robot}, \psi] = [0,0,0]$. Although Fig. 12(a) shows a very consistent map laser scanner – furrows are parallel –, when the robot turns into the following furrow, odometric information is not longer reliable – see Fig. 12(b). Therefore, the map information looses its consistency. The range of the laser used in the experiments shown herein was of 30 m.

Fig. 13 shows a partial map of the olive grove. It corresponds to a portion of Fig. 10(a). The blue triangles correspond to the position

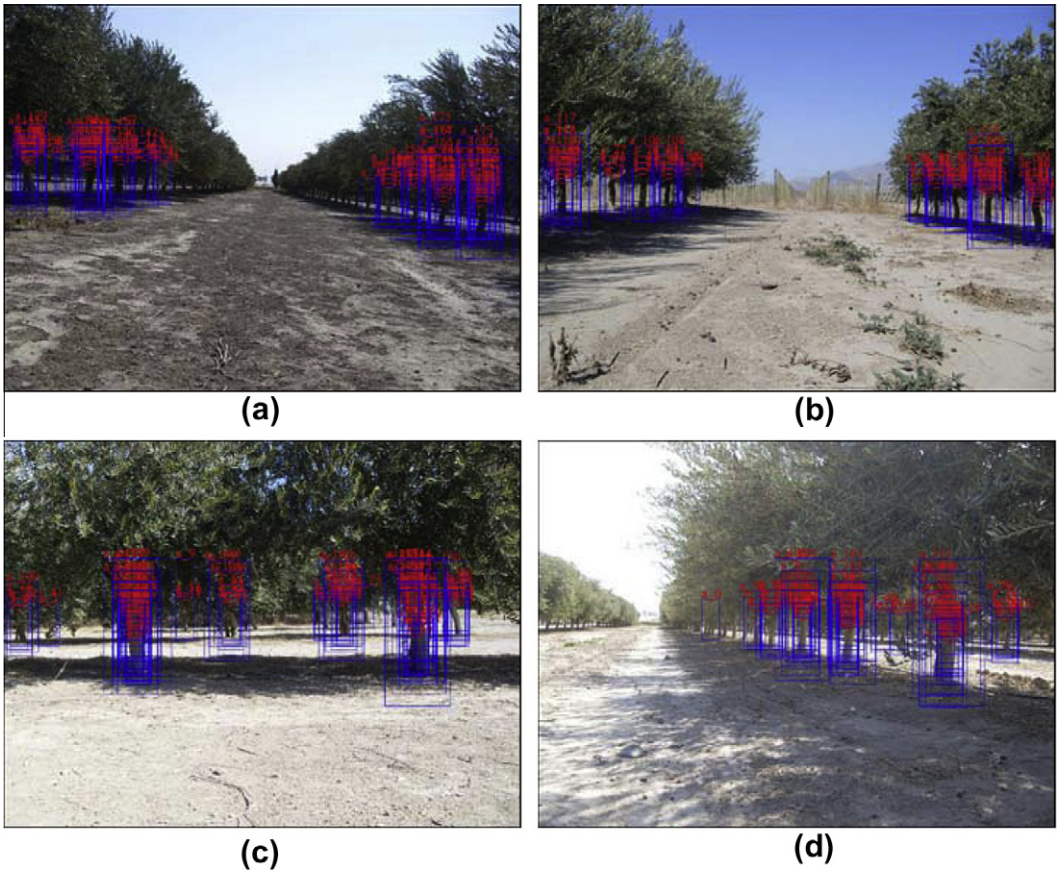

Fig. 11. Four examples of olive stems detection by the monocular vision system presented in this work. The blue rectangles that circumscribe the stems represent the multi-scale stem detection. In red are the labels associated with each scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

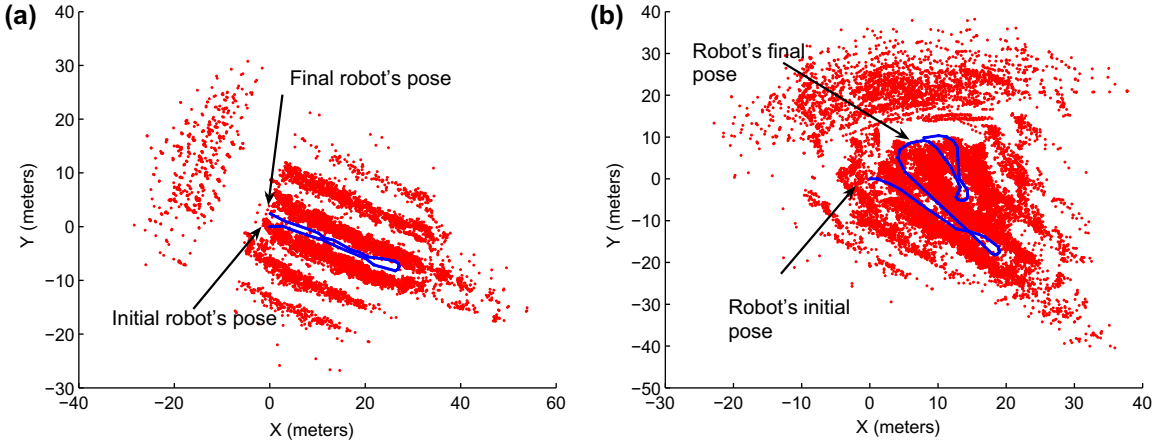

Fig. 12. Odometric-based reconstruction of the olive grove environment. (a) Shows a consistent map laser scanner of the olive grove whereas (b) shows an inconsistent example. The red points correspond to raw laser data and the blue-dotted line is the path travelled by the mobile robot according to the odometric information of the vehicle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

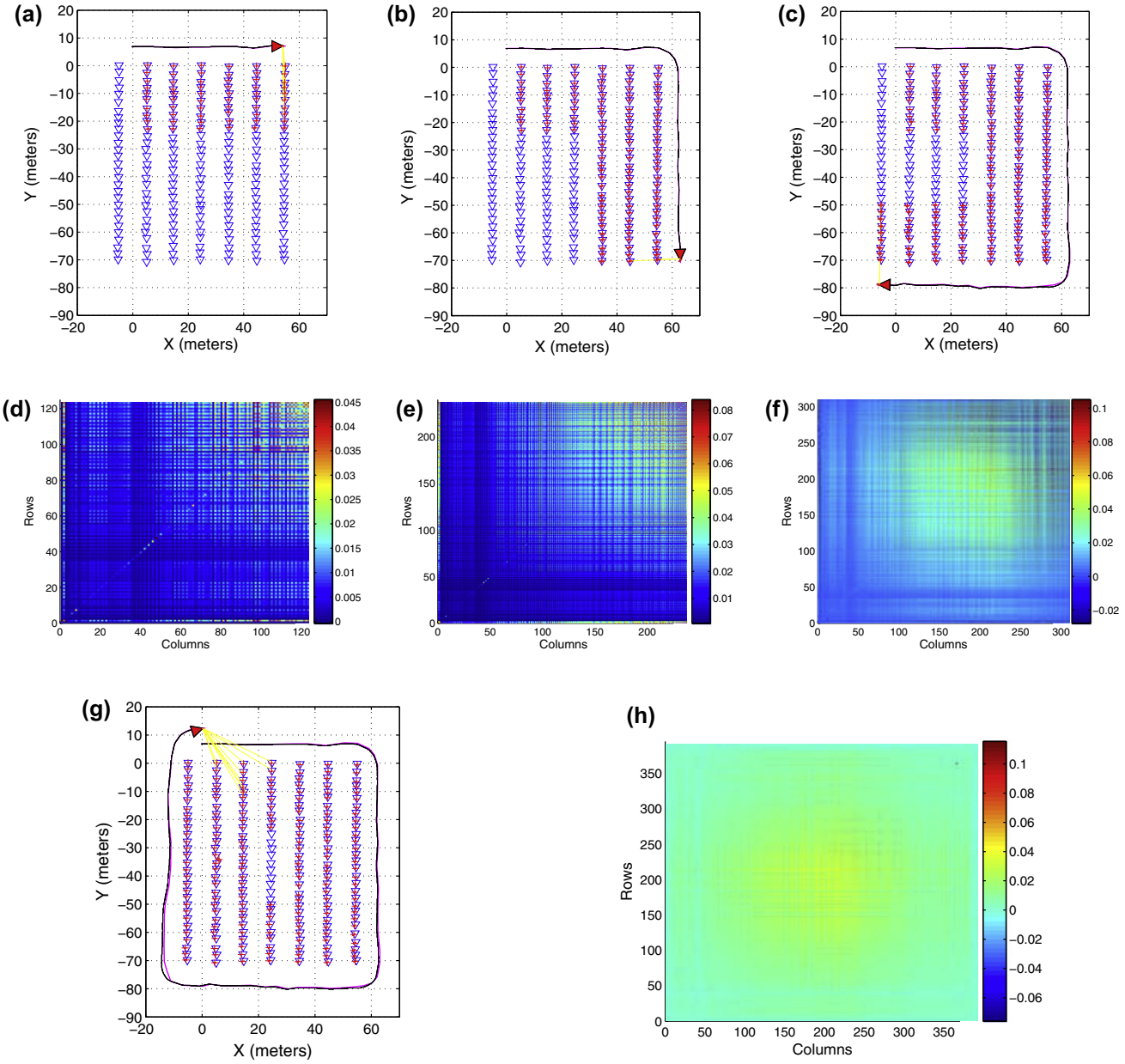

Fig. 13. Partial map reconstruction of the olive grove. Four snap-shoots are shown in this figure. (a) Shows the first snap-shoot and the covariance matrix representation (figure (d) of the EIF-SLAM system state associated with the pose of the mobile robot at the moment of the snap-shoot). (b) and (e) show the second snap-shoot and the corresponding EIF-SLAM covariance matrix representation. (c, f) and (g, h) show the third and the fourth snap-shoot of the experimentation, respectively. The mobile robot is represented in solid-red triangle; the path travelled by the vehicle is shown in black – estimated path – and in magenta – path measured by the differential GPS. The blue triangles are the olive stems of the environment whereas the red crosses represent the olive stems detected by the SLAM algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the trees within the environment. The trees' position were calculated based on differential GPS information regarding the environment. The differential GPS used was built by Novatell with a mean error of 0.2 m. In Fig. 13, the path travelled by the mobile robot and estimated by the EIF-SLAM algorithm is shown in black, whereas in magenta it is shown the path according to the differential GPS measurements. The detected stems from the olive grove are marked by red crosses which also have a covariance ellipse associated with them. The Mobile robot pose is represented by a solid-red triangle. On its front, the mobile robot representation has the covariance ellipse associated with its pose. The robot's initial pose was $[x_{robot}, y_{robot}, \psi] = [-0.3, 6.85, 0]$. The mobile robot's navigation was controlled by a hand-joystick because autonomous navigation problem is not the objective of this work.

Fig. 13 shows four snap-shoots of the environment reconstruction, taken during the mobile robot navigation. Fig. 13(a) shows the mobile robot navigation up to the first corner of the environment. As it can be seen (Thrun et al., 2005) the robot does not re-see the features detected at its initial position. Fig. 13(d) shows this open loop situation. It represents the covariance matrix of the EIF-SLAM system state associated with the estimation process. We show the covariance matrix instead of the information matrix due to the situation shown in Eq. (10) – the elements of the matrix information tend to infinity whereas the covariance matrix entries tend to zero. As it can be seen in Fig. 13(d), the last detected stems have the weakest correlation with the features detected during the initial pose of the robot – see the upper right of Fig. 13(d).

Fig. 13(b) shows the second snap-shoot of the olive grove map reconstruction. As it can be seen, again the loop is not closed and the features from the upper right corner of Fig. 13(b) are the ones with the weakest correlation with respect to the environment mapped so far. Those features are the last detected features and therefore, they are located at the end of the EIF-SLAM system state

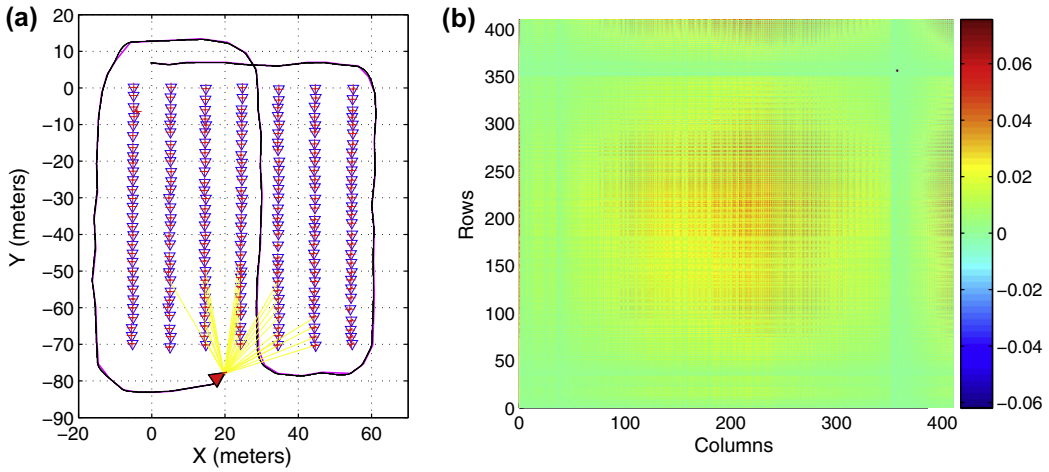

Fig. 14. Full reconstruction of the olive grove. (a) Shows the map built by the SLAM algorithm and the path travelled by the mobile robot. The solid-red triangle represents the robot's final pose; the blue triangles represent the stems of the environment – acquired by means of a differential GPS – and the red crosses show the features detection performed by the SLAM algorithm. The solid black line is the path travelled by the vehicle and estimated by the SLAM algorithm whereas the solid magenta line is the path measured by the differential GPS. (b) Shows a graphical representation of the final covariance matrix of the SLAM system state. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

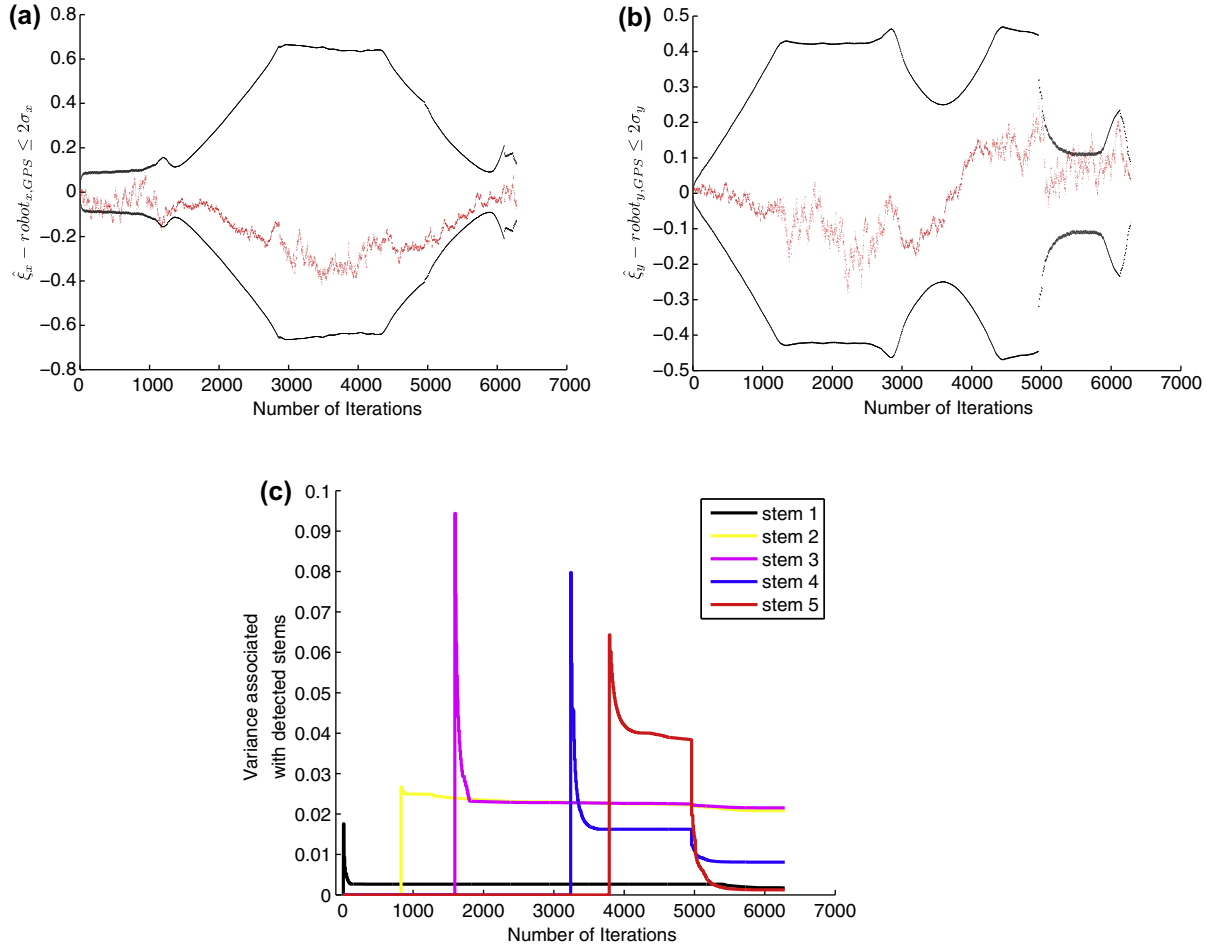

Fig. 15. Consistency analysis of the EIF-SLAM with the features selection criterion proposed in this work. (a) and (b) Show that the robot's error position remains bounded by two time its standard deviation (σ_x and σ_y). (c) Shows the evolution of the variance associated with several stem detected during the SLAM execution of Fig. 13. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

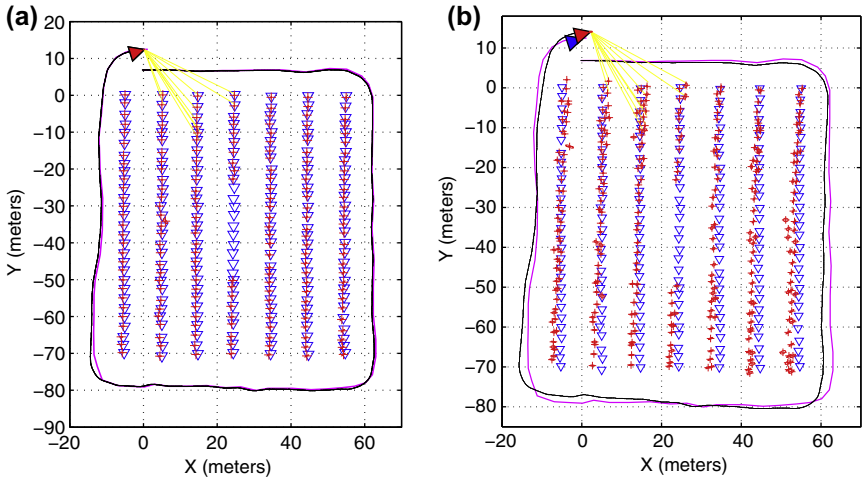

Fig. 16. Maps comparison. (a) Shows the map obtained by the EIF-SLAM with features selection criterion and (b) shows the map build by the EIF-SLAM with no features restrictions. As it can be seen, (a) shows a more coherent map and the estimated features – red crosses – are consistent with the previously mapped ones (by means of differential GPS measurements) – blue triangles –. The solid red triangle is the estimated robot's pose whereas the solid black triangle is the GPS-based robot's pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and at the end of its covariance matrix. Thus, the upper right of Fig. 13(e) shows the weakest correlation with respect to the rest of the features mapped so far.

Fig. 13(c) shows the third snap-shoot of the experiment. As it can be seen, the loop is being closed by the mobile robot navigation. This situation also reflects that the middle of the olive grove has not been mapped by the EIF-SLAM. Therefore, the features near the un-mapped area have the weakest correlation with respect to the rest of the detected features – see Fig. 13(f).

Fig. 13(g) shows the fourth snap-shoot. The navigation loop is finally closed and the first features seen at the initial pose of the vehicle are detected again. As it can be noticed, the middle part of the olive grove remains without mapping, therefore the correlation of the features near that area is weak with respect to the rest of the map. Also, the variance values of the detected features decrease once the loop is closed by the mobile robot – see Fig. 13(h).

Finally, Fig. 14 is included to show how the entire olive grove is constructed by the SLAM algorithm. In Fig. 14, the mobile robot navigates between furrows mapping in this way the features that remained without being observed during the experimentation shown in Fig. 13. Fig. 14(a) shows the map constructed by the EIF-SLAM and Fig. 14(b) shows the graphical representation of the covariance matrix of the SLAM system state. As it can be seen, the covariance matrix of the SLAM process tends to be fully correlated. Therefore, all stems were detected during the navigation and the entire map was built successfully – see Fig. 14(a).

8.2. SLAM consistency

Fig. 15 shows that the unmanned mobile robot localization errors remain bounded by two times their standard deviation. The *number of iterations* refers the number of times that the SLAM algorithm is executed. The consistency test shown in Fig. 15 corresponds to the experiment shown in Fig. 13 above, where the estimated robot's position was extracted from the SLAM system state and the *true* position was the differential GPS measurements ($robot_{GPS}$). On the other hand, Fig. 15(c) shows the evolution of the covariance associated with certain features within the environment when executing the SLAM algorithm of the experiment of Fig. 13. As it can be seen, the variance of the features decreases as the SLAM algorithm is executed.

Fig. 15 also shows a comparison between the SLAM localization procedure and the differential GPS measurements (robot paths in solid black and magenta respectively in Fig. 13). As it can be seen, the error in coordinates x or y of the robot's position does not exceeds 0.5 m in both cases.

In order to see the advantages of EIF-SLAM with features selection criterion, Fig. 16(b) shows the map reconstruction of the environment when no features selection procedure is taken into account. In order to do so, both SLAM algorithms – with features selection and with no features selection – were executed in parallel. The path travelled by the mobile robot and recorded by the full-features SLAM shows an error higher than the one with features selection criterion when compared with the differential GPS-based path. This is due to the fact that as the processing time increases – because of no features selection criterion – and the mobile robot does not stop its motion, the SLAM algorithm misses features from the environment and the loop might not be closed. Also, the map of the SLAM with no features selection criterion is less consistent than the map obtained by the SLAM with features selection restriction incorporated on it, both with respect to the differential GPS trees measurements. Fig. 13 is repeated in Fig. 16(a) for visualization purposes.

As stated above, the sampling time of the EIF-SLAM algorithm with features selection approach was of 1 s, whereas for the full-features case, the maximum time of a single SLAM execution was of 12.2 s (therefore, according to the maximum velocity values presented before, the robot has possibly navigated 2.44 m without extracting features from the environment).

9. Conclusion

In this work, an optimized EIF-SLAM applied to olive groves' environment was presented. The optimization criterion was based on the selection of the most significant features from the agricultural environment according to the information matrix divergence associated with the estimation process. The features extracted from the environment correspond to stems of the olive grove. The stems were extracted by both: a SVM method implemented on a monocular vision-based system and a range laser sensor. Both methods contribute with the range and orientation parameters associated with the stems with respect to the vehicle's position. The agricultural mobile robot used in the experiments is an unmanned car-like vehicle.

The experimental results were carried out at the *National Institute of Technological Agriculture, INTA*, Argentina. The features

extraction procedure has shown to be successful. The range laser sensor and the monocular vision based stems extraction procedures have been implemented in real time, showing that the fusion of both methods produces a robust stem detection. For example, if the monocular vision based system detects a false stem at the middle of a furrow, the laser rejects such detection. Also, if the laser detects a person on the furrow – whose shape morphology can be associated with a stem – it is rejected by the monocular vision system.

In addition, the map reconstruction process based on the most significant features selection criterion of the EIF-SLAM has shown to be consistent with the real environment. As it can be seen in the experimental results section herein, the olive grove reconstruction by the SLAM algorithm, the robot localization and the stems detection have shown a successful performance within the agricultural environment. Also, the consistence of the estimation process was also shown within the graphics of the covariance matrices associated with the EIF-SLAM system state. The covariance matrices were shown instead of the information matrix due to the fact the information matrices – as stated in Eq. (10) – tends to diverge.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.compag.2011.07.007.

References

Auat Cheein, F., di Sciascio, F., Scaglia, G., Carelli, R., 2010. Towards features updating selection based on the covariance matrix of the SLAM system state. Robotica (Cambridge) 29, 271–282.

Ayache, N., Faugeras, O., 1989. Maintaining a representation of the environment of a mobile robot. IEEE Transactions on Robotics and Automation 5, 804–819.

Benton Derrick, J., Bevly, D.M., 2009. Adaptive steering control of a farm tractor with varying yaw rate properties. Journal of Field Robotics 26, 519–536.

Cariou, C., Lenain, R., Thuilot, B., Berducat, M., 2009. Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. Journal of Field Robotics 26, 504–518.

Chatila, R., Laumond, J.P., 1985. Position Referencing and Consistent World Modeling for Mobile Robots. In: Proc. of the IEEE International Conference on Robotics and Automation, pp. 138–145.

Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., Calway, A., 2007. Robust Real-Time Visual SLAM Using Scale Prediction and Exemplar Based Feature Description. IEEE Conference on Computer Vision and Pattern Recognition, 1–7.

Cole, D.M., Newman, P.M., 2006. Using laser range data for 3D SLAM in outdoor environments. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 1556–1563.

Dalal, N., Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893.

Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M., 2001. A solution to the simultaneous localisation and map building (SLAM) problem. IEEE Transactions on Robotics and Automation 7, 229–241.

Durrant-Whyte, H., Bailey, T., 2006a. Simultaneous localization and mapping (SLAM): part I essential algorithms. IEEE Robotics and Automation Magazine 13, 99–108.

Durrant-Whyte, H., Bailey, T., 2006b. Simultaneous localization and mapping (SLAM): part II state of the art. IEEE Robotics and Automation Magazine 13, 108–117.

Flint, A., Mei, C., Reid, I., Murray, D., 2010. Growing semantically meaningful models for visual SLAM. IEEE Conference on Computer Vision and Pattern Recognition, 467–474.

Garulli, A., Giannitrapani, A., Rossi, A., Vicino, A., 2005. Mobile robot SLAM for line-based environment representation. In: IEEE Conference on Decision and Control.

Gonzalez, R.C., Wooks, R.E., 1993. Digital Image Processing. Addison-Wesley Publishing Company.

Guivant, J., Nebot, E., Baiker, S. 2000. High accuracy navigation using laser range sensors in outdoor applications. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3817–3822.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall.

Harville, D.A., 1997. Matrix Algebra from a Statistician's Perspective. Springer-Verlag.

Jin, J., Tang, L., 2009. Corn plant sensing using real-time stereo vision. Journal of Field Robotics 26, 591–608.

Johnson, D.A., Naffin, D.J., Puhalla, J.S., Sanchez, J., Wellington, C.K., 2009. Development and implementation of a team of robotic tractors for autonomous peat moss harvesting. Journal of Field Robotics 26, 549–571.

Lindeberg, T., 1994. Scale-space theory: a basic tool for analysing structures at different scales. Journal of Applied Statistics, 224–270.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110.

Masson, F., Guivant, J., Nieto, J., Nebot, E., 2005. The hybrid metric map: a solution for precision farming. Latin American Applied research 32, 105–110.

Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12, 181–201.

Nagasaka, Y., Saito, H., Tamaki, K., Seki, M., Kobayashi, K., Taniwaki, K., 2009. An autonomous rice transplanter guided by global positioning system and inertial measurement unit. Journal of Field Robotics 26, 537–548.

He, N., Cao, J., Song, L., 2008. Scale Space Histogram of Oriented Gradients for Human Detection. In: International Symposium on Information Science and Engineering, pp. 167–170.

Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H., 2007. 6D SLAM3D mapping outdoor environments. Journal of Field Robotics 27, 699–722.

Schleicher, D., Bergasa, L.M., Ocana, M., Barea, R., Lopez, M.E., 2009. Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion. IEEE Transactions on Intelligent Transportation Systems 10, 440–452.

Thrun, S., Burgard, W., Fox, D., 2005. Probabilistic Robotics. MIT Press.

Vapnik, V.N., 1998. Statistical Learning Theory. Wiley-Interscience.

Zhang, Z., 1999. Flexible camera calibration by viewing a plane from unknown orientations. International Conference on Computer Vision, 666–673.