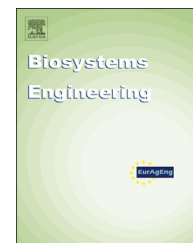


Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

ScienceDirect

journal homepage: [www.elsevier.com/locate/issn/15375110](http://www.elsevier.com/locate/issn/15375110)

## Research Paper

# Image-based particle filtering for navigation in a semi-structured agricultural environment



Santosh Hiremath<sup>a,\*</sup>, Frits K. van Evert<sup>b</sup>, Cajo ter Braak<sup>a</sup>, Alfred Stein<sup>c</sup>,  
Gerie van der Heijden<sup>a</sup>

<sup>a</sup> Biometris, Wageningen University, Wageningen, Netherlands

<sup>b</sup> Plant Science International, Netherlands

<sup>c</sup> Faculty of Geo-Information Science and Earth Observation, University of Twente, Netherlands

## ARTICLE INFO

## Article history:

Received 3 September 2013

Received in revised form

28 January 2014

Accepted 19 February 2014

Published online 19 March 2014

## Keywords:

Vision-based navigation

Row following

Particle filter

Agricultural robots

Autonomous navigation of field robots in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. The drawback of existing systems is the lack of robustness to these uncertainties. In this study we propose a vision-based navigation method to address these problems. The focus is on navigation through a maize field in an outdoor environment where the robot has to navigate through a corridor formed by two plant rows, detect the end of the rows, navigate the headland and turn into another corridor under natural conditions. The method is based on a Particle Filter (PF) using a novel measurement model, where we construct a model image from the particle and compare it directly with the measurement image after elementary processing, such as down-sampling, excessive-green filtering and thresholding. The new measurement model does not extract features from the image and thus does not suffer from errors associated with the feature extraction process. We show how PF can be used for robust navigation of a robot in a semi-structured agricultural environment such as maize fields with inherent uncertainty. We demonstrate the robustness of the algorithm through experiments in several maize fields with different row patterns, varying plant sizes and diverse lighting conditions. To date we have logged over 5 km of successful test runs in which the robot navigates through the corridor without touching the plant stems, accurately detects the end of the rows and traverses the headland.

© 2014 IAGrE. Published by Elsevier Ltd. All rights reserved.

## 1. Introduction

In recent years, there is an increasing demand to reduce the use of chemicals in agriculture due to environmental, economic, and food safety reasons. Potential ways to achieve this are through precision agriculture and organic farming. These

methods, however, require automation of many agricultural operations including harvesting, scouting, tilling, sowing and weed control. Automation is necessary in precision agriculture because the detection of events that might require attention (e.g. weeds, end-of-row, wrong planting density, lack of nutrients, presence of diseases or insects) and the subsequent actuation can be rather time consuming. This

\* Corresponding author.

E-mail addresses: [santoshathiremath@gmail.com](mailto:santoshathiremath@gmail.com) (S. Hiremath), [frits.vanevert@wur.nl](mailto:frits.vanevert@wur.nl) (F.K. van Evert), [cajo.terbraak@wur.nl](mailto:cajo.terbraak@wur.nl) (C. ter Braak), [a.stein@utwente.nl](mailto:a.stein@utwente.nl) (A. Stein), [gerie.vanderheijden@wur.nl](mailto:gerie.vanderheijden@wur.nl) (G. van der Heijden).  
<http://dx.doi.org/10.1016/j.biosystemseng.2014.02.010>

1537-5110/© 2014 IAGrE. Published by Elsevier Ltd. All rights reserved.

makes the process slow and thus labour intensive if a driver has to be present all the time. If a robot is able to complete the task without exploiting any human resources, even time-consuming automatic processes may become affordable. For a machine (or robot) to perform these tasks automatically, it should be able to navigate through the environment autonomously.

The importance of autonomous navigation in agriculture was realised as early as 1958 when Morgan (1958), developed an autonomously navigating tractor. Since then, many different navigational methods, using different sensors, have been developed for agricultural robots. For instance, navigation can be based on leader cables (Tillett & Nybrant, 1990), optical triangulation (Shmulevich, Zeltzer, & Brunfeld, 1989), Global Position System (GPS) (Heidman, Abidine, Upadhyaya, Hills, & Robert, 2002; Slaughter, Giles, & Downey, 2008; Stoll & Kutzbach, 2001), laser-based sensors (Ahamed et al., 2004; Barawid, Mizushima, Ishii, & Noguchi, 2007) and machine vision sensors (van Evert et al., 2011b; Foglia & Reina, 2006; Kondo & Ting, 1998). Good review papers on different autonomous navigating systems in agriculture can be found in various papers (Keicher & Seufert, 2000; Li, Imou, Wakabayashi, & Yokoyama, 2009; Reid, Zhang, Noguchi, & Dickson, 2000; Wilson, 2000).

Autonomous navigation in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. The robot is in a dynamic and non-deterministic environment with many sources of noise. For instance, noise can be due to uneven terrain or varying shapes, sizes and colour of the plants or inconsistency in the environment structure. The problem is further compounded by hardware-related noise like imprecise sensor measurements, wheel-slippage, controller and actuator noise. Designing a navigation method to handle the total uncertainty due to the multiple and often complex sources of variation is not straightforward.

One of the main drawbacks of the existing prototype systems is the lack of robustness to the various sources of uncertainties of the environment (Li et al., 2009). The goal of this study is to develop a vision-based localisation and navigation method based on probabilistic methods to account for the different uncertainties in the environment. We focus on navigation through a maize field in an outdoor environment where the robot has to navigate in the corridor, detect the end of the rows, navigate in the headland and turn into another corridor under natural conditions. We restrict ourselves to vision-based navigation because a camera is the most cost effective and information rich sensor compared to others (Chen, Samarabandu, & Rodrigo, 2007).

The use of probabilistic methods for robot navigation has been subject to extensive studies for the past decade as it provides a good mathematical framework to deal with uncertainty and noisy conditions (Thrun, Burgard, & Fox, 2005). In agriculture, one of the earliest vision-based robots to use probabilistic methods is a horticulture robot that uses the Extended Kalman Filter (EKF) to navigate along the rows of plants (Hague & Tillett, 1996). In the correction step of the EKF, a feature-based measurement model is used, where features are lines extracted using the Hough Transform (Duda & Hart, 1972) from the measurement image. The main drawback of

the system is that it is heavily dependent on the correct extraction of features. Failure of the feature extraction step means that incorrect features are presented to the EKF, injecting noise into the state estimation. Hague and Tillett (1996) minimised this problem by including a filtering step in which highly unlikely features found with the Hough Transform were rejected. Later systems employed in essence similar methods (Åstrand & Baerveldt, 2002; Ericson & Åstrand, 2009; Sogaard & Olsen, 2003; Sogaard & Lund, 2007). Southall, Hague, Marchant, and Buxton (2002) developed a robot to navigate in a field where plants are grown in a grid-like pattern, based on EKF. Using knowledge of the environment they built a grid-based model of the robot's view which is used as the measurement model. One of the shortcomings of this system is the assumption that the local view has a fixed grid structure. This will lead to errors because the grid size and shapes are not often fixed in agricultural fields. Other similar methods include those presented by Marchant (1996) and Marchant and Brivot (1995).

In this study, we propose a novel measurement model to avoid the uncertainty introduced by the feature extraction process in feature-based measurement models such as those that use Hough Transform. We achieve this by constructing a model image (Section 2) from the particle state vector and comparing it directly with the image (after elementary processing, like excessive-green filtering, down-sampling, and barrel distortion correction). As far as we know, this is the first attempt to model the image directly from the particle state vector instead of using a feature extraction process in between. In addition, we use a particle filter (PF) instead of an EKF, primarily because a PF allows a straightforward implementation of the model (albeit at the cost of increased computational load).

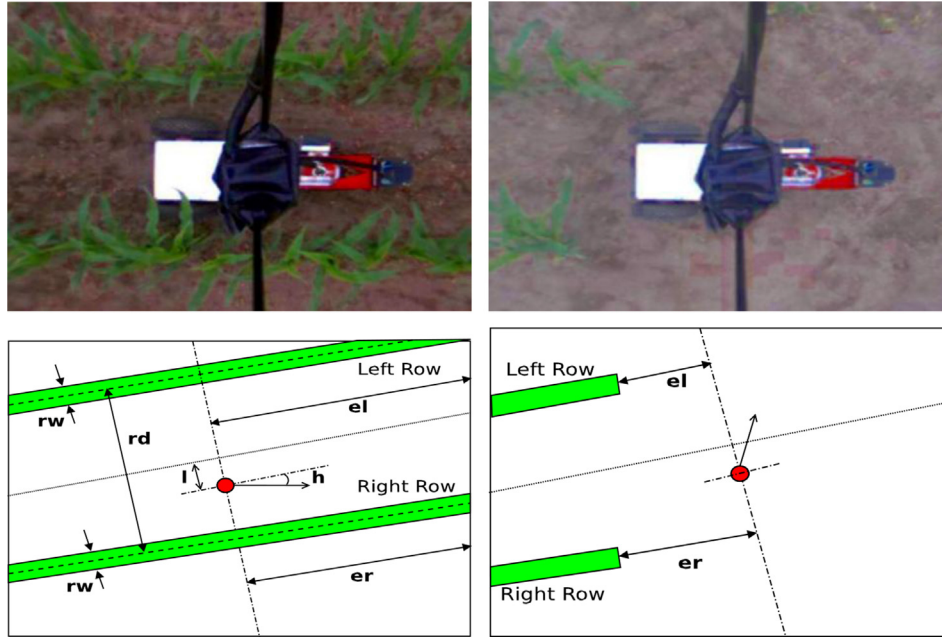
The paper is arranged as follows. Section 2 describes image-based particle filter along with the details of the new measurement model. In Section 3 we demonstrate the performance and robustness of the new navigation algorithm in various operating conditions followed by the discussion section where we point out limitations and extensions of the method and the scope for future research.

## 2. Image-based particle filter

We use a small robot which has to navigate through a field of maize. The robot has a downward-looking camera at a height of 1.65 m, through which it senses the world. The task of the robot is to navigate through a corridor formed by two adjacent rows of maize plants, detect the end of the rows, navigate in the headland and return into another row under natural conditions. This can be divided into two distinct situations: navigation between the rows and navigation on the headland. Figure 1a shows a typical camera view when the robot is between the rows and Fig. 1b shows a typical camera view when the robot is on the headland.

### 2.1. Local world

Our local world is a field with maize plants that consists of two parallel rows of equal width ( $rw$ ), with a distance ( $rd$ ) between



**Fig. 1** – Typical camera views when the robot is in between the rows (top left) and on headlands (top right). Figures in the bottom row illustrate the parameterisation of the local world. The robot control point is represented by the solid circle and its travelling direction is indicated by the arrow attached to it:  $h$  is the robot heading,  $l$  is the lateral deviation,  $rd$  is the distance between the rows,  $rw$  is the row width,  $el$  is the end of left row, and  $er$  is the end of right row.

them. The end of the rows is indicated by  $el$  for the left row and  $er$  for the right row. The  $rw$ ,  $rd$ ,  $el$  and  $er$  are measured in metres. The robot is characterised by its main axis (direction of travel straight ahead) and the control point (CP) halfway between the wheels. The position of the robot in this world is parameterised by the robot heading ( $h$ ) and lateral deviation ( $l$ ). The heading is the angle between the robot's main axis and the reference axis (line along the centre of the rows) and is measured in radians. Lateral deviation is the perpendicular distance between the robot and the reference axis measured in metres. The parameters together represent the robot–environment state at time  $t$  and is denoted by  $X_t = (h, l, rw, rd, el, er)$ .

The state of the robot is estimated by a PF. The PF maintains a posterior distribution over  $X_t$  given the past sensor measurements  $Z_{1:t}$  and the past controls  $U_{1:t}$ . This posterior distribution is approximated by a set of samples called particles. A particle is a possible realisation of  $X_t$ . The posterior probability  $p(X_t|Z_{1:t}, U_{1:t})$  at time  $t$  can be expressed in terms of that at time  $t - 1$  in the formula

$$p(X_{1:t}|Z_{1:t}, U_{1:t}) = \frac{p(Z_t|X_t)p(X_t|X_{t-1}, U_t)}{p(Z_t|Z_{1:t-1})} p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1}) \quad (1)$$

where,  $p(Z_t|X_t)$  represents the measurement update step and is given by the measurement model (also called the likelihood model),  $p(X_t|X_{t-1}, U_t)$  represents the motion update step and is given by the motion model,  $p(Z_t|Z_{1:t-1})$  is the normalising constant, and  $p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1})$  is the posterior distribution at previous time step  $t - 1$ . Details of the measurement model are discussed below. The specific form of (1) is indicative of the recursive nature of a particle filter where the posterior probability at the previous time step is updated by

multiplying it by the motion model and the measurement model obtained from the current time step. A brief description of the particle filter algorithm is included in the next section.

When the robot is between the rows, the navigation algorithm steers the robot to follow the reference line along the centre of the rows. At the row ends, it stops 0.6 m outside the rows, makes a 90° turn and aims to follow a line parallel to the ends of the rows (van Evert et al., 2011a).

## 2.2. Particle filter

The particle filter algorithm is shown in Table 1. The set of  $m$  samples representing the posterior distribution is denoted by where each particle is a concrete instantiation of the state of the system at time  $t$ . Lines 2–6 correspond to the motion update step where new particles are sampled based on the state transition probability  $p(X_t|X_{t-1}, U_t)$ . Subsequently, the weight  $w_t$  of each particle is calculated based on the measurement probability  $P(Z_t|X_t)$ . The weight  $w_t$  is an indication of how well the predicted particle ‘agrees’ with the measurement  $Z_t$ . The predicted particles along with their weights constitute the temporary particle set  $\hat{X}_t$ . These particles are resampled based on their weights to get  $X_t$  that represents the belief at time  $t$  (lines 7–10). The resampling method used is importance sampling which consists of drawing, with replacement,  $m$  particles from the set  $\hat{X}_t$  with each particle having a probability for being included in the new set  $X_t$  proportional to its weight (Thrun et al., 2005). The resampling step ensures that the particles with high weight – which best represent the current state of the system – are more likely to be retained while those with low weight are filtered out.

**Table 1 – Particle filter algorithm.**

1. Particle Filter ( $X_{t-1}, U_t, Z_t$ )
2. for  $k = 1$  to  $m$  do
3. sample  $X_t^{(k)} \sim P(X_t | X_{t-1}^{(k)}, U_t)$
4.  $w_t^{(k)} = P(Z_t | X_t^{(k)})$
5. add  $(X_t^{(k)}, w_t^{(k)})$  to  $\bar{X}_t$
6. end for
7. for  $k = 1$  to  $M$  do
8. draw  $i$  with probability  $\propto w_t^{(i)}$
9. add  $X_t^{(i)}$  from  $\bar{X}_t$  to  $X_t$
10. end for
11. return  $X_t$

### 2.3. Motion model

The motion model describes how the state vector changes from one time step to another. When the robot is between the rows, it should follow the path along the centre of the corridor and when travelling on headland, it should follow the path along an imaginary line parallel to the line connecting the row ends. In both cases, the robot aims to follow a specified reference line. We assume that the distribution of the initial state ( $X_0$  at  $t = 0$ ) of the robot is known, and that the gyroscope and wheel encoders on the robot provide the control information  $U_t = (dx, dh)$  where  $dx$  is the displacement of the robot along its heading and  $dh$  is the turning angle of the robot. Now, the motion model is given by

$$\begin{aligned}
 h_t &= h_{t-1} + dh + \varepsilon_h \\
 l_t &= l_{t-1} + dx \sin(h_{t-1} + dh) + \varepsilon_l \\
 rw_t &= rw_{t-1} + \varepsilon_{rw} \\
 rd_t &= rd_{t-1} + \varepsilon_{rd} \\
 el_t &= el_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{el} \\
 er_t &= er_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{er}
 \end{aligned} \tag{2}$$

where  $\varepsilon_h, \varepsilon_l, \varepsilon_{rw}, \varepsilon_{rd}, \varepsilon_{el}, \varepsilon_{er}$  are independent Gaussian noise applied to the corresponding state variables.

A complication arises because the end of the rows is frequently not in view of the robot's camera. According to the motion model, the values of  $el$  and  $er$  are constantly decreased. When the end-of-row is not in robot's view, the  $el$  and  $er$  values should not be decreased. This situation is dealt with by re-initialising  $el$  and  $er$  in a fraction of the particles at every update.

### 2.4. Measurement model

The measurement  $Z_t = (z_1, z_2, \dots, z_m)$  at time  $t$  is a binary image of size  $m = r \times c$  pixels. Pixels for which the value is one ( $z_i = 1$ ) are mostly green and indicate the probable presence of plant material; pixels for which the value is zero ( $z_i = 0$ ) indicate the probable absence of plant material. The binary image  $Z_t$  is obtained from the camera image after preliminary image processing. First, the RGB image from the camera is down-sampled to  $r \times c$  pixels, then it is converted to an excessive-green image  $I = (-R + 2G - B)/3$  which is thresholded to yield the binary image.

The measurement model for  $Z_t$  depends on a model image. The model image is a discretised version of the local world as parameterised by the particle and has the same number of

pixels as the measurement image. If we assume that the pixels in  $Z_t$  are independent and Bernoulli-distributed with parameters  $Q = (q_1, q_2, \dots, q_m)$ , then the measurement model is given by the likelihood

$$p(Z_t | X_t; Q) = \prod_i q_i^{z_i} (1 - q_i)^{1-z_i} \tag{3}$$

with  $q_i$  being the probability that pixel  $i$  is green.

As our local world consists of plant rows and soil, pixels in the model image belong either to a plant row (in-row region I) or to soil (out-row region O). A parsimonious model is obtained by setting  $q_i = \theta_I$  when the pixel belongs to the in-row region and  $q_i = \theta_O$  when it falls in the out-row region where  $\theta_I > \theta_O$  as we expect more green in in-row region relative to out-row region. The likelihood can then be rewritten as

$$p(Z_t | X_t; \Theta) = \prod_{(r \in \{I, O\})} \prod_{i \in r} \theta_r^{z_i} (1 - \theta_r)^{1-z_i} \tag{4}$$

where  $\Theta = \{\theta_I, \theta_O\}$ .

### 2.5. Fuzzy membership

Hard clipping the excessive-green image to a binary image heavily depends on the threshold value used. This may lead to large differences in the binary image, under only slightly changing lighting conditions. To avoid this effect, we propose to use soft clipping over a range  $[a, b]$  instead of hard clipping. This implies that  $z_i$  is no longer a binomial variable but a pseudo-binomial variable and it is calculated as follows. Let  $I_i$  be the excessive-green value, which is transformed to a value  $z_i$  between 0 and 1 as follows.

$$z_i = \begin{cases} 0 & : I_i < a \\ 1 & : I_i > b \\ \frac{I_i - a}{b - a} & : a \leq I_i \leq b \end{cases}$$

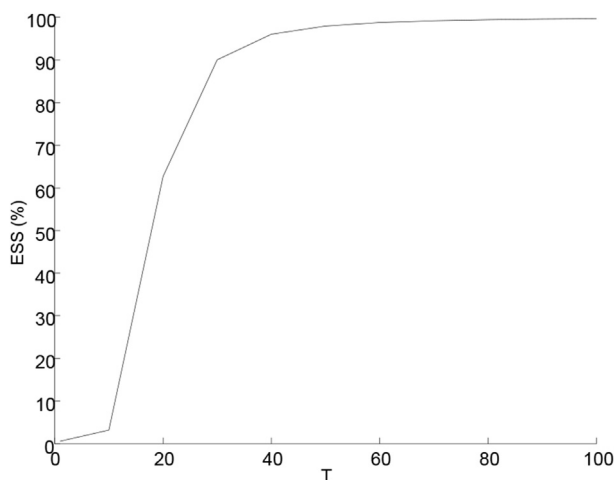
Note that  $z$  is no longer a binary variable any more, but can assume values between 0 and 1. Hence the distribution now becomes pseudo-binomial instead of binomial. This is according to the theory for analysis of fractions as described by e.g. McCullagh and Nelder (1989). The equations (3) and (4) are identical for the pseudo-binomial case. The values  $a$  and  $b$  can be determined empirically using images taken at different lighting condition and examining the threshold value at which the segmentation is successful.



## 2.6. Attenuation factor

Within the PF framework the likelihood model determines the weight of each particle obtained from the motion model. After computing the weights, the particles are resampled according to their weights to yield the posterior distribution. The peakedness of the likelihood model is a measure of the diversity of the posterior distribution. A highly peaked likelihood means less diversity of particles. The likelihood as described in Section 2.4 is highly peaked because the dependency between the pixels is not considered. Pixels in the image are dependent due to correlation prevalent in the local world. For instance, pixels are dependent because they belong to the same plant and there may be correlation at larger distances due to the regularity in their growth pattern. In the measurement model, however, every pixel is regarded as an independent measurement. This leads to a large number of measurements resulting in a highly peaked likelihood where the likelihoods of two similar particles differ by orders of magnitude. For example, for an image of size  $40 \times 60$  pixels, if two particles have a difference of 0.04 m (one pixel) in lateral deviation, their likelihoods may differ by up to a factor of  $3.37 \times 10^{216}$  when  $\theta_l = 0.8$  and  $\theta_o = 0.02$ . This peakedness causes particle degeneracy in the PF. This problem is addressed by expressing the dependency between the pixels by means of the attenuation factor  $T$  (Thrun et al., 2005). Instead of counting every pixel as an individual measurement,  $T$  pixels are considered as a single measurement and reduce the number of measurements by a factor  $T$  by raising the likelihood to the power  $1/T$ .

The appropriate value of  $T$  can be determined using the effective sample size (ESS) of the PF (Doucet & Johansen, 2009). Figure 2 shows the plot of ESS against  $T$  averaged over the iterations of a PF run. As can be seen at small values of  $T$ , the ESS is close to 1% indicating most particles have negligible weights. But as the value of  $T$  is increased, the ESS increases indicating that the weights are more evenly distributed among the particles. We selected the value of  $T$



**Fig. 2 – The effective sample size (ESS) as a function of the attenuation factor,  $T$ , where likelihood is tempered by taking the  $T$ th root. As  $T$  increases, effective sample size also increases indicating a smoother likelihood function.**

such that 80% of the particles are retained after the resample step. We also observe that the curve is not smooth. This is because the coarse resolution of the values of  $T$  used in the experiment.

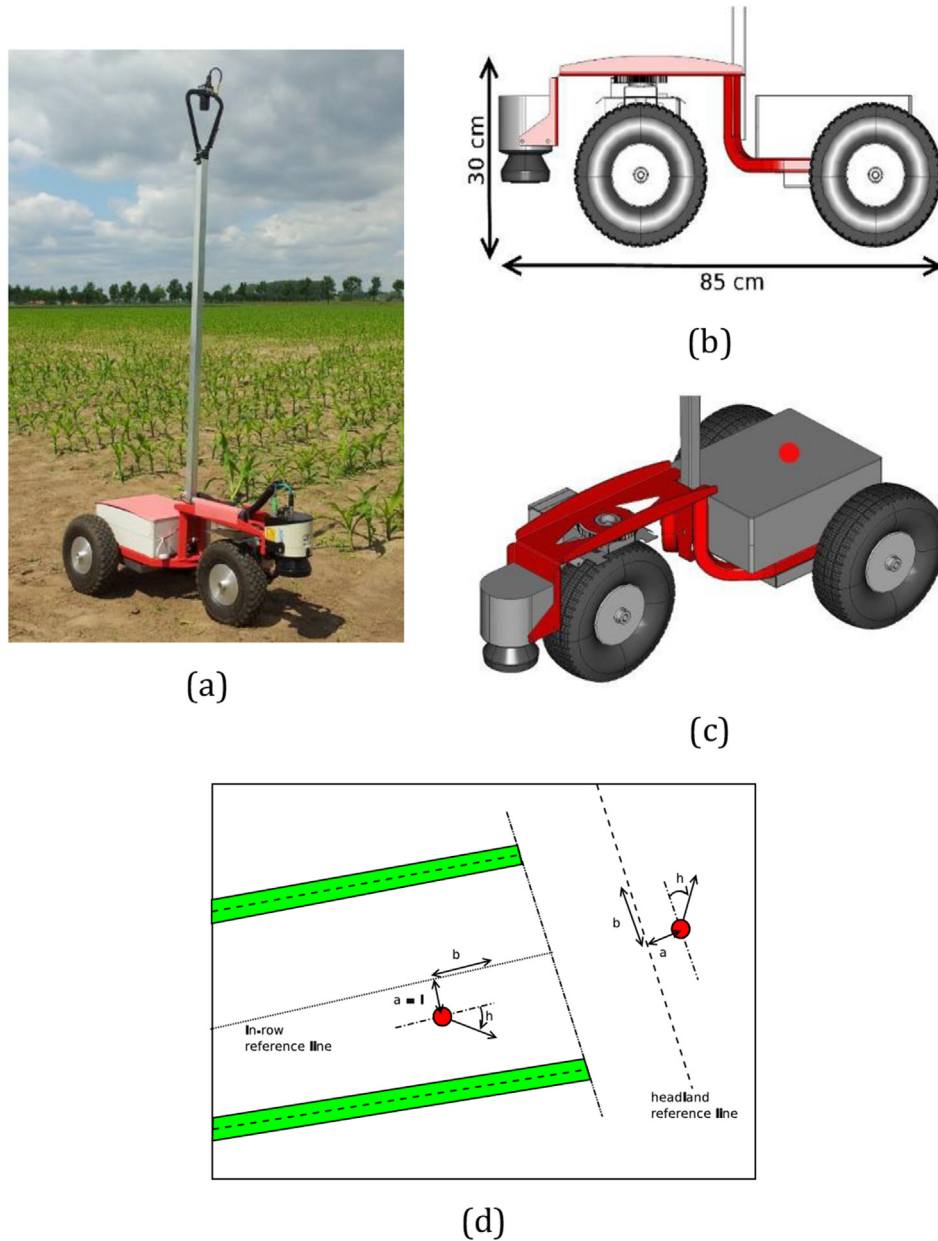
## 3. Experiments and results

### 3.1. Robot

The robot employed in this study consists of a chassis with three wheels, with overall dimensions  $0.8 \text{ m} \times 0.45 \text{ m} \times 0.3 \text{ m}$ . It has an actuated front wheel as the steering wheel that is affected by commands from a control program by means of a CAN-bus and it has two rear wheels that do not have the ability to steer. All wheel units are equipped with incremental encoders to measure the rotational speed. In addition, the front wheel unit is equipped with an angle sensor to measure the steering angle. The driving speed of each wheel depends upon the target speed of the control point, the location of the wheel with respect to the control point and the turning radius. An electronic box between the rear wheels houses a mini-ITX computer with a 2.4 GHz Intel Core2 Duo processor running Windows XP operating system. The robot is controlled by a custom C# software which uses OpenCV library for image processing. Energy to the computer and the wheel units is provided by three 12 V NiMH racing packs: one for the front wheel unit, one for both rear wheel units, and one for the PC.

The wheel units are equipped with incremental encoders to measure the rotational speed of each wheel. The controller steers the robot by trying to follow a predetermined path pattern. A simple string notation is used to indicate the path pattern. For example the string '1L-2R' is an instruction to navigate to the end of the corridor and turn into the adjacent corridor on the left, navigate to the end and then turn into the second corridor to the right. When the robot is in the corridor, it follows the in-row reference line and when it is in the headland it follows the headland reference line. In either case the target steering angle of the front wheel unit is given by  $\gamma = -h + \tan^{-1}(-a/b)$  where  $h$  is the robot heading,  $a$  is the perpendicular distance of the robot from the reference line and  $b$  is the target distance along the reference line as indicated in Fig. 3d. Note that  $a = l$  if the robot is between the rows. After detecting the end of the plant rows, the robot continues to follow the rows until its control point is at a given distance beyond the end of the rows. It then makes an on-the-spot turn to position itself parallel to the headland reference line and continues the headland navigation where  $\gamma$  is now calculated with respect to headland reference line. Upon reaching the middle of the next corridor (new target corridor according to string pattern), it comes to a full stop, turns to position itself along the middle and starts following the new in-row reference line which is along the middle of the new corridor.

The in-row reference line is shifted so as to be along the middle of the corridor along which the robot is travelling. An element of navigation on the headland is counting how many rows have been crossed in order to follow the string pattern. To do so, a row counter is used that is initially set to zero and is



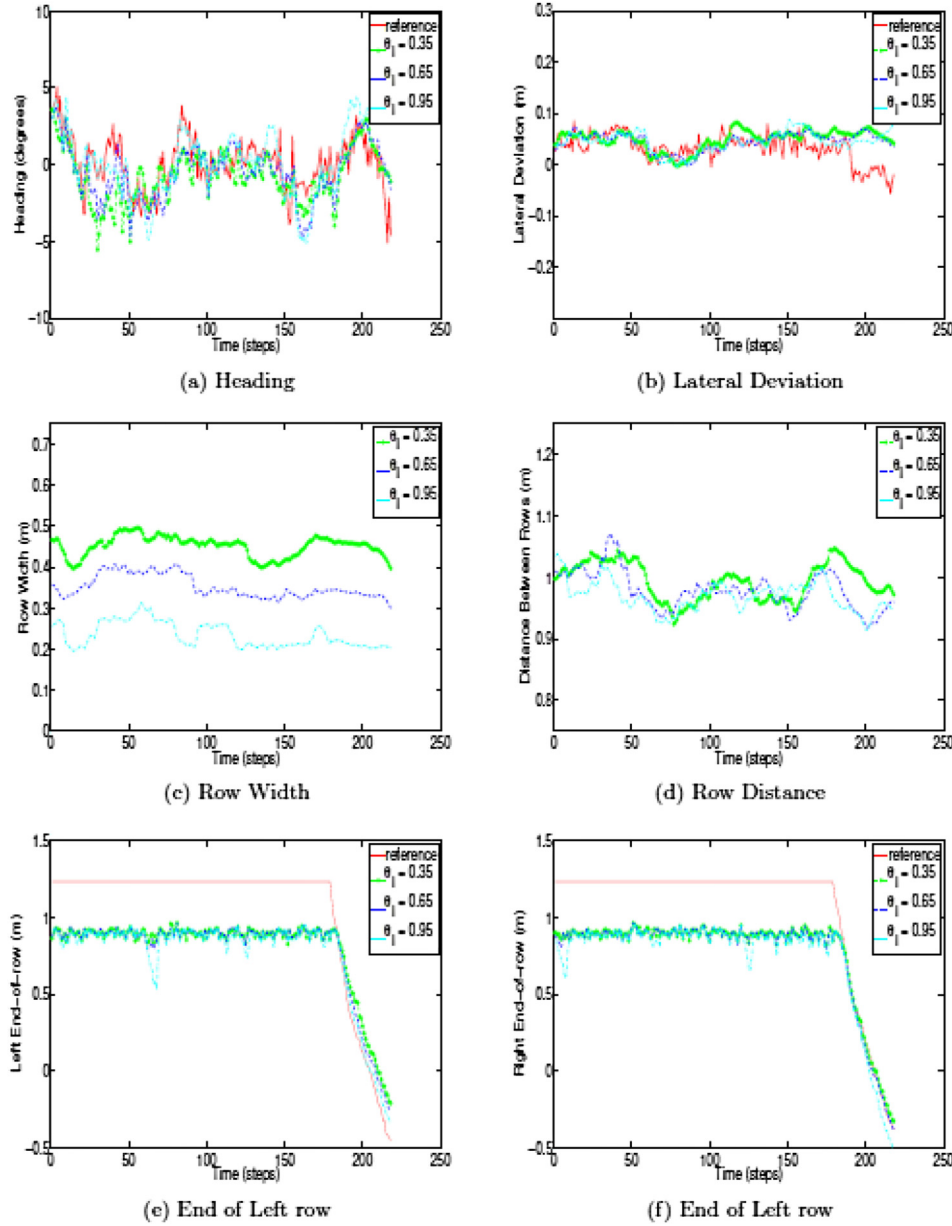
**Fig. 3 – (a) The robot with the mounted camera in a field. (b) The profile view of the robot depicting the length and height of the robot drawn to scale. (c) Perspective view of the robot showing the electronics box between the rear wheels. The red dot between the two rear wheels is the designated control point that indicates the position of the robot in the environment. (d) The target steering angle  $\gamma$  as it is computed by the controller. Between rows the robot follows the in-row reference line (dotted line) and within the headland the robot follows the headland reference line (dashed line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)**

updated when the lateral deviation ( $l$ ) of the robot is larger than half the row distance ( $rd$ ).

The robot is equipped with a downward-looking camera (uEye UI-1220, IDS Imaging Development Systems GmbH, Obersulm, Germany) at a height of 1.65 m, through which it senses the world. The camera captures an RGB image of size width  $\times$  height = 752  $\times$  480 pixels where height is in the travelling direction and width is perpendicular to it. The input image is subjected to preliminary processing to obtain the measurement image as described in Section 2.4.

### 3.2. Calibration

The experimental data consists of several sequences of images taken by the robot's camera as it navigates through the field. The sequences are from different maize fields with different plant size, row structure — straight rows, curved rows, gaps within the rows — and under different lighting conditions. The number of images in each sequence depends on the length of the robot run in which the camera records data at 10 Hz. For every image in the sequence, the ground



**Fig. 4** – The result of particle filter estimation of  $X_t$  of a robot run, for three values of  $\theta_l$ . The dotted curve with circular marker shows the particle filter estimate with  $\theta_l = 0.35$ , the dashed curve shows the particle filter estimate when  $\theta_l = 0.65$ , and the dashed and dotted curve shows the particle filter estimate when  $\theta_l = 0.95$ . The value of  $\theta_o$  was fixed to 0.02 in all three cases. The solid line in red indicates the reference data.

truth for the most important state variables, heading ( $h_r$ ), lateral deviation ( $l_r$ ), left end-of-row ( $el_r$ ) and right end-of-row ( $er_r$ ) was established by means of hand annotation. These ground-truth values were used to evaluate the performance of the PF.

Experiments consisted of running the PF with several image sequences for various values of  $\theta_l$  and  $\theta_o$ . In all the experiments, the PF was initialised with 256 particles. This number is empirically determined based on real-time computational constraints – a balance between update frequency (10 Hz) and number of particles. The particles are initialised as follows:  $h$  has a uniform distribution in the range  $[-10, 10]^\circ$ ,  $l$  has a uniform distribution in the range  $[-0.1, 0.1]$  m,

$rw$  has a uniform distribution in the range  $[0.05, 0.6]$  m,  $rd$  has a uniform distribution in the range  $[0.5, 1.5]$  m and  $el$  and  $er$  have a uniform distribution in range  $[1.2, 1.4]$  m. During the update step, each component of the state vector is updated based on the motion model where the error components are sampled from the following distributions:  $e_h \sim N(0; 1.0) (^\circ)$ ,  $e_l \sim N(0; 0.01)$  (m),  $e_{rw} \sim N(0; 0.01)$  (m),  $e_{rd} \sim N(0; 0.01)$  (m),  $e_{el} \sim N(0; 0.02)$  and  $e_{er} \sim N(0; 0.02)$  (m). The particles are resampled according to importance sampling (Thrun et al., 2005).

After the resample step, the weighted mean of the posterior distribution  $\hat{X}_t$  is the PF estimate of the state  $X_t$  at time  $t$ . Thus, for every image in the sequence, its PF estimate can be compared with the corresponding ground truth. Figure 4

shows the result of one such comparison for  $\theta_o = 0.02$  and three different values of  $\theta_l$  in  $\{0.35, 0.65, 0.95\}$ . The six diagrams in the figure correspond to the six state variables. Variables  $h$ ,  $l$ ,  $el$  and  $er$  are plotted along with their corresponding ground-truth data  $h_r$ ,  $l_r$ ,  $el_r$  and  $er_r$ , respectively, while  $rw$  and  $rd$  have no reference.

As we see in Fig. 4a, the PF successfully tracks  $h_r$  except when the robot reaches the end of the rows where it diverges. In the case of lateral deviation, the PF tracks  $l_r$  throughout the run (Fig. 4b). When the end of the rows is in sight, the PF is able to detect this and tracks them as shown in Fig. 4e and f, but when the end of the rows is not in sight, the values are of course not correct.

We also observe that the estimation of  $h$ ,  $l$ ,  $er$ ,  $el$  and  $rd$  is not sensitive to a large range of values of  $\theta_l$ . This is in contrast with  $rw$  (Fig. 4c) which is inversely proportional to  $\theta_l$ . Similar effects were observed for different values of  $\theta_o$  (not illustrated).

The algorithm performs well over a broad range of values of  $\Theta = \{\theta_l, \theta_o\}$  and its actual value is thus not critical. By setting the parameter halfway in the range we expected it to perform well over the broadest range of circumstances. You can nicely see the compensating effect of the row width with the fraction  $\Theta$  used. A lower  $\Theta$  results in a wider row, including soil (O) pixels of the outer-row region. Hence the fraction automatically adjusts itself. This exchangeability is one of the reasons not to estimate both fraction and row width in the same go, but to fix one of them.

The model was tested with five independent image sequences. Table 2 gives the mean and standard deviations of the Root Mean Squared Error of these five test runs. The tests were carried out with  $\theta_l = 0.65$ ,  $\theta_o = 0.02$ . The values show that the state vector is estimated with a high degree of precision.

### 3.3. Field tests

The field in which the robot was tested consisted of rows of maize plants with a well-defined headland. These rows may be either straight or curved. Additionally, there may be gaps in the rows because of missing plants. The rows are spaced 0.75 m apart, being the distance at which the seeding took place. Figure 5 shows four example images of the various field conditions in which experiments were conducted. The first row of images shows fields with various row patterns in which experiments were carried out. The second and third rows of images show the lighting conditions and plant sizes under which the test was conducted. This includes low light causing poor visibility, bright sunlight causing shadows, and plant sizes ranging from 10 cm to 50 cm.

A test run consisted of two laps of a 60 m long path alternating through the five rows. The robot started at the

beginning of the first corridor, travelled along the plant rows, exited into the headland on the other end of the corridor, travelled along the headland to the next corridor and turned into it and so on. Figure 6 shows the trajectory that the robot followed during the test runs. Each row is approximately 12.6 m long while the headlands are about 7 m. The total distance covered by the robot in a single test run is 140 m.

To date we have recorded a cumulative distance of more than 5 km in various field conditions. The path of a test run and its length were designed to ensure maximum variability in navigation scenarios encountered by the robot. For instance, the robot has to navigate the headland in two different scenarios: one with plants on its right and the other with plants on its left. Similarly, the robot encounters scenarios where it has to exit or enter the rows from both right and left. It also tests the effects of shadows due to sunlight and changed plant orientations due to wind direction. Table 3 shows the test runs conducted over the years 2011–2012 at different times of the day. In all cases the robot successfully traversed the specified path. That is, it navigates without touching the stems of the plants while in the corridor, accurately detects the end-of-row, traverses the headland at the specified distance from the row ends, and turns automatically at the requested corridor.

The navigation method was also tested for its computational speed and real-time performance. Concerning the speed, the algorithm processes up to 10 images per second. This image rate allowed the robot to successfully navigate the field at a speed up to  $0.7 \text{ m s}^{-1}$  ( $2.5 \text{ km h}^{-1}$ ). The computational speed of the algorithm is also influenced by the size of the image, which should be small enough to ensure real-time operability of the algorithm. Therefore the camera image is downsized to retain only the essential information needed to accurately determine the orientation of the rows. Due to the difference in the information content when the robot is in the row and in the headland, different image sizes are used for the two situations. Particularly, when the robot is in the row the image is  $60 \times 47$  pixels covering an area of  $2 \text{ m} \times 1.5 \text{ m}$ . Here 2 m is along the direction of the travel and 1.5 m perpendicular to it. In the headland the image is  $90 \times 141$  pixels and covers an area of  $2 \text{ m} \times 3 \text{ m}$  where 2 m is along the direction of the travel and 3 m is perpendicular to it. Increasing the image size in the headland ensures that sufficient information is captured to estimate the robot position while keeping the computational speed within the real-time operating limit. The speed on the headland is half the speed in the corridor.

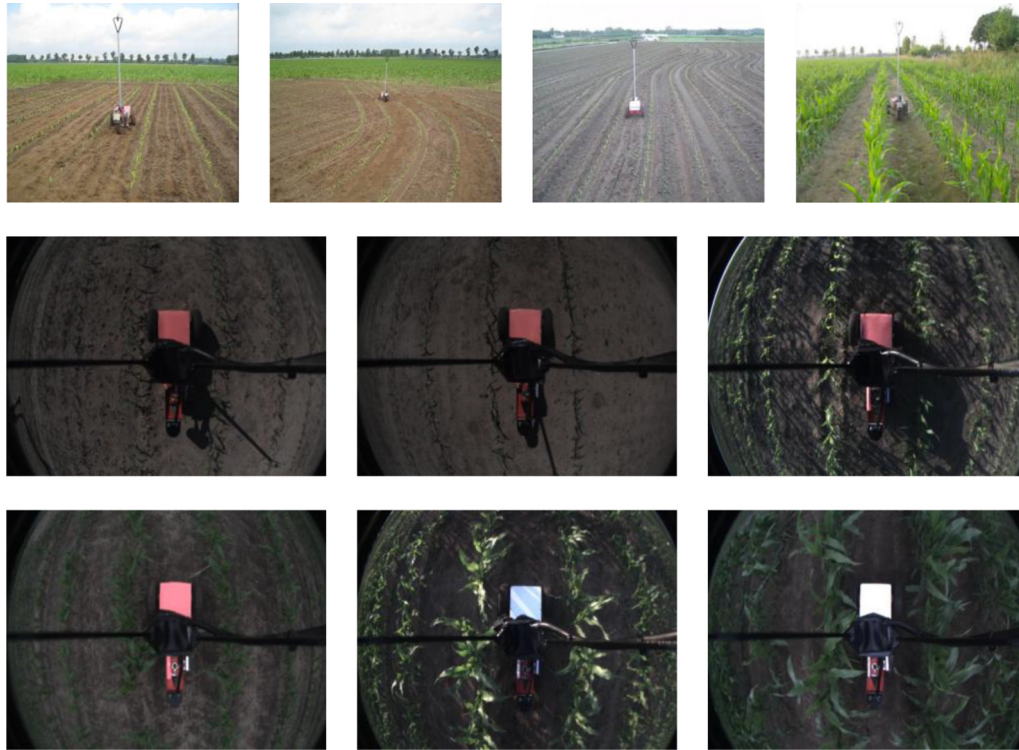
## 4. Discussion

Experiments showed that estimation of all components of the state vector except  $rw$  is robust to changes in  $\theta_l$  and  $\theta_o$ . The value of  $rw$  is increased when  $\theta_l$  is large in comparison with the plant cover. The influence of values  $\theta_l$  and  $\theta_o$  is noticeable only when plants are small or plant density is low. In such situations, extreme values like  $\theta_l = 0.95$  lead to erroneous estimates of  $rw$ , which eventually leads to the failure of the algorithm. This, however, is easily prevented by a rough calibration or estimation of  $\theta_l$  according to the specific stage of plant growth. This allows the possibility of using different

**Table 2 – Mean  $\pm$  standard deviation of the Root Mean Squared Error of five test runs.**

Heading (degrees)	$3 \pm 1$
Lateral deviation (m)	$0.04 \pm 0.007$
Left end-of-row (m)	$0.22 \pm 0.04$
Right end-of-row (m)	$0.24 \pm 0.05$



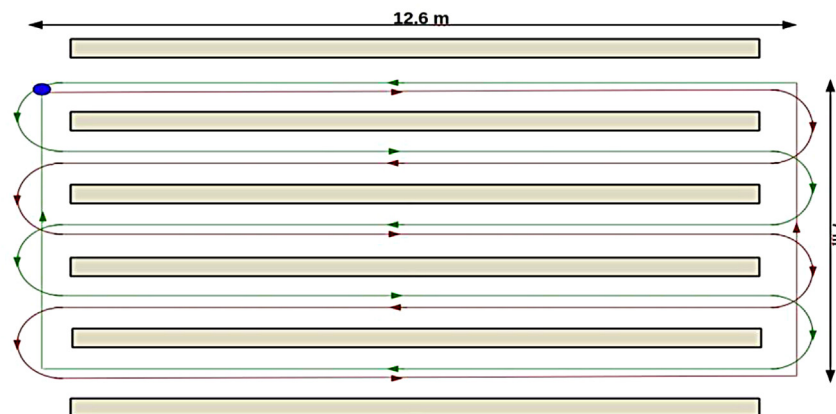


**Fig. 5 – (Top row) Example of maize fields in which the robot was tested. (Second and third row) Example images show the different lighting conditions in which the experiments were conducted along with various plants of various sizes. The poor quality of some of the images in the second and the third row illustrate the extremity of the field conditions in which the tests were conducted.**

values for young plants (less than 5 cm in height) and older plants (greater than 50 cm in height) as confirmed in the field experiments. The results showed that, in the case of young plants, the algorithm works best with the values  $\theta_1$  in the range [0.2,0.4]. Similarly, with older plants the best values of  $\theta_1$  were in the range [0.6,0.8]. It is worth noting that the value of  $\theta_1$  also influences the sensitivity of the algorithm to detect the end-of-row. Using high values of  $\theta_1$  makes the algorithm more sensitive to end of the rows as shown by the dashed and

dotted curve in Fig. 4e between time steps 60 and 75 where the value of  $el$  decreases due to a gap in the row. This behaviour is advantageous as it can be exploited to detect gaps in the rows if necessary by adjusting the value of  $\theta_1$ .

The estimation of the state vector is less accurate during headland following compared to row following. This is due to the relative lack of information in the headland. When the robot is in the headland only a small part of the rows is visible making the estimation of the orientation of the robot heading



**Fig. 6 – The trajectory followed by the robot during the test runs. The blue circle indicates the starting position. For illustrative purposes, the red and green paths do not run exactly halfway between the rows.**

**Table 3 – Number of test runs during the period 2011–2012 at different times. The tests runs were distributed across several days in a month. This distribution pattern was not systematic as it does not affect the experiment. Each test run consists of two laps, however, some test cases were restricted to a single lap due to insufficient battery power to operate the robot.**

Date/Time	9:00–12:00	12:00–18:00	18:00–21:00
June, 2011	—	9.5	7.5
May, 2012	4.5	—	—
June, 2012	2	2.5	7.5
July, 2012	1	1	0.5

with respect to the rows less precise. This, however, did not lead to any mistakes by the robot and it was rectified automatically as the robot turned into the new row.

The heading and lateral deviation parameters are generally unimodal and could well be described by a Gaussian distribution. Therefore we could have used a Kalman Filter as well for these parameters. The distribution for end-of-row parameters will be multimodal when there are gaps in the rows. Since these gaps are all indicators of potential end-of-row there may be multiple peaks in the distribution that need to be tracked and a non-parametric filter like PF can address this in a straightforward manner.

The assumption that the rows have equal width is based on the observation that the width of the two rows are highly correlated and adding a separate variable for the width of each row mainly adds to the complexity of the model. Incorrect estimates of row width and row distance will result in a different number of foreground and background pixels in the two regions (in-row/out-row). As long as the fraction in the in-row and out-row region are substantially different, the Bernoulli likelihood works well, even if  $\theta$  is rather far from the optimum for these two regions. A smaller  $\theta$  for in-row generally leads to an increased estimate of row width since more background pixels will then be included in the in-row region. Thus the row width parameter has a compensating effect for an incorrect setting of the fraction  $\theta$ . Apart from substantially wrong settings of  $\theta$  for in-row and out-row, the row width parameters can well be estimated, as is demonstrated by the experiments with plants of different ages where the row width varies significantly.

The downsizing factor of the image is a trade-off between the computational constraints and the accuracy of the algorithm. It was determined empirically to ensure that the robot could process an image every 5–10 cm, which resulted in a speed of  $0.7 \text{ m s}^{-1}$  and an image processing frequency of 10 Hz. The downsizing factor has an influence on the attenuation factor for reducing the peakedness of the likelihood model. The peakedness of the likelihood model is an artefact of the assumption that each pixel is an independent measurement although it is clear that the adjacent pixels are correlated. Taking the Tth root overrides the independence assumption by grouping T pixels as a single measurement. A more rigorous approach would be to relax the independence assumption and explicitly model the correlation between the pixels (e.g. using a Markov Random Fields). This would also provide a more realistic representation of the plant structure.

The camera is set to auto exposure and auto white balancing. This takes care of variation of the green in different illumination situations. The formula for excessive green is motivated based on the colour contrast between plant and soil. It is well known in the agricultural literature that excessive green is a rather invariant measure for distinguishing between plant and soil and it is used in many applications (Golzarian, Lee, & Desbiolles, 2012).

While developing the robotic system, we have used and tested numerous approaches, starting with a Hough Transform and Kalman Filter. Occasionally the Hough Transform yielded wrong line estimates, which made the robot behave unreliably. Also the bimodality of the end-of-row parameter forced us to look for alternatives for the standard Kalman Filter/Hough Transform. Using a Particle Filter with image generation instead of a Kalman Filter and Hough Transform improved the end-of-row detection and yielded a reliable robotic navigation system.

Constructing a model image from a state vector to directly define the likelihood of the particles is a novel way to circumvent the difficulties and uncertainties associated with any feature extraction process. While it is preferable in environments with sparse features with simple geometric representation, it can be challenging in a complex environment with a high degree of uncertainty from many sources of noise. We believe this approach has clear advantages over feature extraction as it circumvents the errors of the feature extraction process. Extending the image-based measurement model to more complex situations may be a challenging field of research.

## 5. Conclusion

In this paper we present a navigation algorithm for navigating a robot in a semi-structured agricultural environment such as a maize field based on a particle filter. We show that the particle filter provides accurate estimates of the robot–environment state accounting for the uncertainty inherent in the environment. By using an image-based likelihood model we mitigate the effects of uncertainty of the feature extraction process accompanying feature-based likelihood models. The new likelihood model is also generic and can be extended to include multiple regions in the image. Peakedness of the likelihood may result from this image-based approach, but can be circumvented by using an attenuation factor based on effective sample size. We demonstrate the robustness of the algorithm through experiments in several fields with different row patterns, varying plant sizes and diverse lighting conditions. To date we have logged over 5 km of successful test runs in which the robot traverses the specified path staying along the middle of the corridor when travelling between the rows, accurately detecting the end of the rows and traversing the headland.

## Acknowledgements

We thank Mr. Martijn van der Bijl and colleagues of Kverneland Group, Nieuw Vennep, The Netherlands, for providing us with the robot, including the microcontroller (low-level

control) software. We are indebted to Mr. Arjan Lamaker, Wageningen, The Netherlands, for writing the PC (highlevel control) software.

## REFERENCES

- Ahamed, T., Takigawa, T., Koike, M., Honma, T., Yoda, A., Hasegawa, H., et al. (2004). Characterization of laser range finder for in-field navigation of autonomous tractor. In *Proceedings of the international conference on automation technology for off-road equipment* (pp. 120–130). Kyoto, Japan.
- Åstrand, B., & Baerveldt, A.-J. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots*, 13, 21–35.
- Barawid, O., Jr., Mizushima, A., Ishii, K., & Noguchi, N. (2007). Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosystems Engineering*, 96, 139–149.
- Chen, Z., Samarabandu, J., & Rodrigo, R. (2007). Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics*, 21, 233–265.
- Doucet, A., & Johansen, A. M. (2009). *The Oxford handbook of nonlinear filtering. Chapter A: Tutorial on particle filtering and smoothing: Fifteen years later*. Oxford University Press.
- Duda, R., & Hart, P. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15, 11–15.
- Ericson, S., & Åstrand, B. (2009). A vision-guided mobile robot for precision agriculture. In *Precision Agriculture'09: Papers presented at the 7th European conference on precision agriculture, Wageningen, The Netherlands, 6–8 July 2009*. Wageningen: Wageningen Academic Publishers.
- van Evert, F. K., van der Bijl, M., Lamaker, A., Stravers, T., Polder, G., van der Heijden, G. W., et al. (2011a). Hugo. In *Proceedings of the 8th Field Robot Event, 2010* (pp. 88–99). Braunschweig, Germany. Available online at <http://www.digibib.tu-bs.de/?docid=00041345>.
- van Evert, F. K., Samsom, J., Polder, G., Vijn, M., van Dooren, H. J., Lamaker, A., et al. (2011b). A robot to detect and control broad-leaved dock (*Rumex obtusifolius* L.) in grassland. *Journal of Field Robotics*, 28, 264–277.
- Foglia, M. M., & Reina, G. (2006). Agricultural robot radicchio harvesting. *Journal of Field Robotics*, 23, 363–377.
- Golzarian, M., Lee, M., & Desbiolles, J. (2012). Evaluation of color indices for improved segmentation of plant images. *Transactions of the ASAE*, 55, 261–273.
- Hague, T., & Tillett, N. (1996). Navigation and control of an autonomous horticultural robot. *Mechatronics*, 6, 165–180.
- Heidman, B., Abidine, A., Upadhyaya, S., Hills, D., & Robert, P. (2002). Application of RTK GPS based auto-guidance system in agricultural production. In *Proceedings of the 6th international conference on precision agriculture and other precision resources management* (pp. 1205–1214). Minneapolis, USA.
- Keicher, R., & Seufert, H. (2000). Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture*, 25, 169–194.
- Kondo, N., & Ting, K. C. (1998). *Robotics for bioproduction systems*. American Society of Agricultural Engineers (ASAE).
- Li, M., Imou, K., Wakabayashi, K., & Yokoyama, S. (2009). Review of research on agricultural vehicle autonomous guidance. *International Journal of Agricultural and Biological Engineering*, 2, 1–16.
- McCullagh, P., & Nelder, J. (1989). *Generalized linear models* (Vol. 37). Chapman & Hall/CRC.
- Marchant, J. (1996). Tracking of row structure in three crops using image analysis. *Computers and Electronics in Agriculture*, 15, 161–179.
- Marchant, J., & Brivot, R. (1995). Real-time tracking of plant rows using a Hough transform. *Real-Time Imaging*, 1, 363–371.
- Morgan, K. (1958). A step towards an automatic tractor. *Farm Mech*, 10, 440–441.
- Reid, J., Zhang, Q., Noguchi, N., & Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, 25, 155–167.
- Shmulevich, I., Zeltzer, G., & Brunfeld, A. (1989). Laser scanning method for guidance of field machinery. *Transactions of the ASAE*, 32, 425–430.
- Slaughter, D., Giles, D., & Downey, D. (2008). Autonomous robotic weed control systems: a review. *Computers and Electronics in Agriculture*, 61, 63–78.
- Søgaard, H. T., & Lund, I. (2007). Application accuracy of a machine vision controlled robotic micro-dosing system. *Biosystems Engineering*, 96, 315–322.
- Søgaard, H. T., & Olsen, H. J. (2003). Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 38, 141–158.
- Southall, B., Hague, T., Marchant, J. A., & Buxton, B. (2002). An autonomous crop treatment robot: Part I. A Kalman filter model for localization and crop/weed classification. *International Journal of Robotics Research*, 21, 61–74.
- Stoll, A., & Kutzbach, H. (2001). Guidance of a forage harvester with GPS. *Precision Agriculture*, 2, 281–291.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics (Intelligent robotics and autonomous agents)*. The MIT Press.
- Tillett, N., & Nybrant, T. (1990). Leader cable guidance of an experimental field gantry. *Journal of Agricultural Engineering Research*, 45, 253–267.
- Wilson, J. (2000). Guidance of agricultural vehicles – a historical perspective. *Computers and Electronics in Agriculture*, 25, 3–9.